目　录

第1章　偏微分方程

偏微分方程在现代科学中扮演了重要角色。偏微分方程是从基本自然规律、应用数学、数学物理以及工程技术中产生的。很多重要的物理问题可以用偏微分方程及相应的初始条件，边界条件或初始-边界混合条件描述。为了使读者能深入地研究偏微分方程及其在计算机视觉中的应用，本章首先引入偏微分方程一些基本概念。进而，给出了二阶线性偏微分方程分类的方法。简要介绍了非线性偏微分方程基本概念，经典的线性和非线性偏微分方程，以及变分原理和极值原理。

1.1　线性偏微分方程

函数 $u(x,y,\cdots)$ 的偏微分方程(Partial Differential Equation)是指函数 u 与其偏导数 $\frac{\partial u}{\partial x}$，$\frac{\partial u}{\partial y}$，$\frac{\partial^2 u}{\partial x^2}$，$\frac{\partial^2 u}{\partial y^2}$，$\frac{\partial^2 u}{\partial x\partial y}$，$\cdots$ 的一个数学关系式，并可以记为

$$F\left(x,y,u,\frac{\partial u}{\partial x},\frac{\partial u}{\partial y},\frac{\partial^2 u}{\partial x^2},\frac{\partial^2 u}{\partial y^2},\frac{\partial^2 u}{\partial x\partial y},\cdots\right)=0。\tag{1.1.1}$$

式中：F 是函数；$x,y,\cdots$ 是自变量；而 $u(x,y,\cdots)$ 是应变量。

偏微分方程的阶次(order)与常微分方程的定义相似。方程(1.1.1)中阶次最高的偏导数次数称该偏微分方程的阶次。有两个自变量 x,y 的一阶(first-order)偏微分方程可以写成

$$F\left(x,y,u,\frac{\partial u}{\partial x},\frac{\partial u}{\partial y}\right)=0。\tag{1.1.2}$$

同样，二阶(second-order)偏微分方程可以写成

$$F\left(x,y,u,\frac{\partial u}{\partial x},\frac{\partial u}{\partial y},\frac{\partial^2 u}{\partial x^2},\frac{\partial^2 u}{\partial y^2},\frac{\partial^2 u}{\partial x\partial y}\right)=0。\tag{1.1.3}$$

以此类推，可得高阶偏微分方程。例如

$$x\frac{\partial u}{\partial x}+y\frac{\partial u}{\partial y}=0,\tag{1.1.4}$$

$$x\frac{\partial u}{\partial x}+y\frac{\partial u}{\partial y}=x^2+y^2,\tag{1.1.5}$$

$$u\frac{\partial u}{\partial x}+\frac{\partial u}{\partial t}=u,\tag{1.1.6}$$

$$\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2=1\tag{1.1.7}$$

是一阶方程，而

$$\frac{\partial^2 u}{\partial x^2}+2\frac{\partial^2 u}{\partial x\partial y}+\frac{\partial^2 u}{\partial y^2}=0,\tag{1.1.8}$$

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0, \tag{1.1.9}$$

$$\frac{\partial^2 u}{\partial t^2}-c^2\frac{\partial^2 u}{\partial x^2}=f(x,y) \tag{1.1.10}$$

是二阶方程，最后

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+\frac{\partial^3 u}{\partial x^3}=0, \tag{1.1.11}$$

$$\frac{\partial^2 u}{\partial t^2}+\frac{\partial^4 u}{\partial x^4}=0 \tag{1.1.12}$$

分别是三阶和四阶方程。

如果偏微分方程中的所有未知函数及其导数之间的关系是线性的，且其系数仅与自变量有关，则称该方程为线性的(linear)。例如，方程(1.1.4)、(1.1.5)，方程(1.1.8)～(1.1.10)和方程(1.1.12)是线性的。

为了方便，可以将偏微分方程写成算子形式：

$$L_x u(x)=f(x)。 \tag{1.1.13}$$

式中：L_x 是算子。如果，对于任意两个函数 u 和 v 以及任意两个常数 a 和 b，算子 L_x 满足

$$L_x(au+bv)=aL_x u+bL_x v, \tag{1.1.14}$$

则称为线性算子(linear operator)。

如果方程(1.1.13)中 L_x 是线性算子，则该方程是线性的。方程(1.1.13)是非齐次(inhomogeneous)线性偏微分方程。如果方程(1.1.13)中 $f(x)\equiv 0$，则称为齐次(homogeneous)方程。方程(1.1.4)，方程(1.1.8)、(1.1.9)和方程(1.1.12)是线性齐次方程，而(1.1.5)，(1.1.10)是线性非齐次方程。

同理，如果方程(1.1.13)中算子 L_x 是非线性的，则该方程是非线性方程。方程(1.1.6)、(1.1.7)和方程(1.1.11)是非线性方程。

方程(1.1.1)的解(solution)是使方程(1.1.1)成立，并且函数 $u=u(x,y,\cdots)$ 在区域 D 内连续可微(differentiable)，也就是函数 u 在区域 D 内偏导数都存在，且偏导数连续。

某些情况下，解在区域 D 内连续可微的条件可以放宽。如果函数 u 或它的偏导数在区域 D 内某些点或所有点上是不连续的，则称方程(1.1.1)的解 $u=u(x,y,\cdots)$ 为弱解(weak solution)或广义解(generalized solution)。

为了引入偏微分方程通解的定义，考察偏微分方程

$$\frac{\partial^2 u}{\partial x\partial y}=0 \tag{1.1.15}$$

的解 $u=u(x,y)$。

将方程(1.1.15)对 x 进行积分，得到

$$\frac{\partial u}{\partial y}=h(y),$$

其中：$h(y)$是 y 的任意函数。再将它对 y 积分，得到

$$u(x,y)=\int h(y)\mathrm{d}y+f(x),$$

其中：$f(x)$是任意函数。或等效为

$$u(x,y)=f(x)+g(y), \tag{1.1.16}$$

其中：$f(x)$和$g(y)$是任意函数。解(1.1.16)称为二阶偏微分方程(1.1.15)的通解。

通常，偏微分方程的通解由任意函数的数学表达式组成。一个简单的方程(1.1.15)可以有无穷多解。例如

$$u(x,t)=f(x-ct)+g(x+ct), \tag{1.1.17}$$

其中：f和g分别是$(x-ct)$和$(x+ct)$的任意函数，那么，

$$\frac{\partial^2 u}{\partial x^2}=f''(x-ct)+g''(x+ct),$$

$$\frac{\partial^2 u}{\partial t^2}=c^2 f''(x-ct)+c^2 g''(x+ct)=c^2\frac{\partial^2 u}{\partial x^2}。$$

式中函数右上角的符号代表求导。这个二阶线性方程称为波动方程

$$\frac{\partial^2 u}{\partial t^2}-c^2\frac{\partial^2 u}{\partial x^2}=0。 \tag{1.1.18}$$

式(1.1.17)定义的函数$u(x,y)$满足方程(1.1.18)，与函数$f(x-ct)$和$f(x+ct)$的具体形式无关，仅要求函数f和g至少是二次可微的函数。因此，式(1.1.17)给出的方程(1.1.18)的通解包含任意函数。

仅有两个独立变量x和y的情况下，方程(1.1.1)的解$u(x,y)$可看成是一个几何曲面，称为(x,y,u)空间中的一个积分面。

1.2　线性叠加原理

n阶线性齐次常微分方程组的通解是n个线性独立解与n个任意常数的线性组合。换言之，如果$u_1(x),u_2(x),\cdots,u_n(x)$是$n$阶线性齐次常微分方程

$$Lu(x)=0 \tag{1.2.1}$$

n个线性独立的解，那么，对于任意常数$c_1,c_2,\cdots,c_n$有

$$u(x)=\sum_{k=1}^{n}c_k u_k(x)。 \tag{1.2.2}$$

上式是方程(1.2.1)的通解，称为常微分方程的线性叠加原理(Superposition Principle)。注意到，通解(1.2.2)严格依赖于n个任意常数。

但对于线性齐次偏微分方程

$$L_x u(\boldsymbol{x})=0, \tag{1.2.3}$$

通解依赖于任意函数而不是任意常数。所以，方程(1.2.3)有无穷多个解。如将方程(1.2.3)无穷多个解的集用$u_1(\boldsymbol{x}),u_2(\boldsymbol{x}),\cdots,u_n(\boldsymbol{x}),\cdots$表示，那么有无穷多个线性组合

$$u(\boldsymbol{x})=\sum_{n=1}^{\infty}c_n u_n(\boldsymbol{x})。 \tag{1.2.4}$$

式中：c_n是任意常数。由于无穷级数常不收敛，式(1.2.4)通常不再是方程(1.2.3)的解。因此，对于偏微分方程，叠加原理可能不成立。但是，如果偏微分方程(1.2.3)仅存在有限数目的解$u_1(\boldsymbol{x}),u_2(\boldsymbol{x}),\cdots,u_n(\boldsymbol{x})$，那么

$$u(\boldsymbol{x})=\sum_{n=1}^{n}c_n u_n(\boldsymbol{x}) \tag{1.2.5}$$

仍是方程(1.2.3)的一个解。如果无穷级数(1.2.4)是收敛的，且级数可以逐项施加算子L_x，

则叠加原理可用于线性齐次偏微分方程，而式 $u(x)$ 是方程(1.2.3)的解。

通常使用分离变量法导出解 $u(x)$。分离变量法与解的叠加原理一起，称为富氏法。

无穷线性组合的另一个应用是寻找给定偏微分方程的解。这涉及到一族依赖于连续实参数 k 和函数 $c(k)$ 的解 $u(x,k)$ 使

$$\int_a^b c(k)u(x,k)\mathrm{d}k \quad 或 \quad \int_{-\infty}^{\infty} c(k)u(x,k)\mathrm{d}k \tag{1.2.6}$$

收敛。那么，在某些条件下，这个积分也是一个解。这也称为线性叠加原理。

线性叠加原理是求偏微分方程解析解的一个重要理论基础。关于这方面的知识，有兴趣的读者可阅读本章参考文献[1]或[5]。

1.3 初始条件与边界条件

大多数情况下，偏微分方程的通解含有任意函数，有一定的不确定性，因此意义不大。必须将通解与其他一些附加条件，如初始(initial)条件或边界(boundary)条件一起考虑才能得到感兴趣的特定解。

初始条件和边界条件是根据特定物理问题的要求而提出的。具有独立时间变量 t 的偏微分方程中，初始条件给出了应变量 $u(x,t)$ 在特定时间 $t=t_0$ 或 $t=0$ 的物理状态。一般情况下，给出初始条件 $u(x,0)$ 或同时给出初始条件 $u(x,0)$ 和边界条件 $u_t(x,t)$ 就可以决定函数 $u(x,t)$ 在以后时间的变化。这类条件称为 Cauchy 条件或初始条件。它常是唯一解存在的必要和充分条件。根据初始条件求偏微分方程解的问题称为 Cauchy 问题或初始值问题(initial-value problem)。

实际上很多物理问题，就是给定空间中区域 D 及其边界 ∂D 上的应变量 $u(x,t)$，在区域 D 中求偏微分方程的解。而且，边界并不一定要求是闭合的有限面积或体积的域。部分边界可以在无穷远处。当然，在这种情况下必须给定无穷远处的边界条件。这类问题称为边界值问题(boundary-value problem)，它是应用数学和物理学中最基本的问题。在物理问题中最常见的有三类边界条件：

(1) Dirichlet 条件：在区域 D 的边界 ∂D 上每个点给定 u 的值，求区域 D 中方程 $L_x u(x)=0$ 的解称 Dirichlet 边界值问题。

(2) Neumann 条件：在区域 D 的边界 ∂D 的每个点上给定法向导数 $\frac{\partial u}{\partial n}$ 的值。对应的问题称为 Neumann 边界值问题。

(3) Robin 条件：边界 ∂D 的每个点上给定 $\left(\frac{\partial u}{\partial n}+au\right)$ 的值。对应的问题称 Robin 边界值问题。

由偏微分方程描述的问题，并在指定区域中给出初始条件和/或边界条件(或其他补充条件)，如能满足下列要求：

(1) 存在性(existence)：这个偏微分方程至少存在一个解。

(2) 唯一性(uniqueness)：偏微分方程最多存在一个解。

(3) 稳定性(stability)：解必须稳定，就是说，给定数据中发生微小变化，其相应解的变化必须保持在允许的范围内。

稳定性指标是求解物理问题的一个基本要求。一个数学方程只有求得其稳定的解，才能探讨该方程实际意义。

1.4 经典线性偏微分方程

介绍偏微分方程，有必要介绍一些重要的线性方程模型。因为对这些方程数学界和物理界已经有了长时期深入研究，形成了一系列成熟的数学处理方法。而且这些模型有助于读者对偏微分方程的物理意义有更深入的了解。本文首先介绍二阶偏微分方程，这是基于下列考虑：第一，二阶偏微分方程具有广泛的实用意义；第二，二阶偏微分方程在数学处理上比较简单，也容易理解。

[例 1.4.1] 下列方程称为波动方程：

$$\frac{\partial^2 u}{\partial t^2} - c^2 \nabla^2 u = 0, \tag{1.4.1}$$

$$\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}。 \tag{1.4.2}$$

c 是常数。这个方程描述了波的扩散(或扰动)。它可以描述很多物理问题，例如，弦的振动，薄膜的振动，杆或梁的纵向弹性振动，水的浅表波动，声学以及电信号在电缆中的传输等问题。

[例 1.4.2] 下列方程称为热扩散方程：

$$\frac{\partial u}{\partial t} - \kappa \nabla^2 u = 0, \tag{1.4.3}$$

其中：κ 是导热系数。式(1.4.3)描述了某种量子的流动，例如，热或一团基本粒子的流动，它在生物学中也被用作描述生长和扩散的过程，特别是肿瘤的生长。这个热扩散方程还可描述在 Stoke 和 Rayleigh 问题中的非稳定附面层流动以及由旋涡面产生的旋涡扩散。

[例 1.4.3] 下列方程称为拉普拉斯方程：

$$\nabla^2 u = 0。 \tag{1.4.4}$$

此方程用来描述无源静电场的电位，引力场，弹性薄膜的平衡位移，不可压缩流体的速度场，稳态热传导问题的温度分布和其他诸多物理现象。

[例 1.4.4] 以下方程称为泊松方程：

$$\nabla^2 u = f(x,y,z), \tag{1.4.5}$$

其中：$f(x,y,z)$ 是一个描述场源或漏的给定函数。这是非齐次拉普拉斯方程。泊松方程表示有源或漏的情况下拉普拉斯方程描述的现象。

[例 1.4.5] Helmholtz 方程为

$$\nabla^2 u + \lambda u = 0, \tag{1.4.6}$$

其中：λ 是常数，此方程就是与时间独立的波动方程(1.4.1)加了一个参数 λ。在声学问题中，它的解代表了一种声音的辐射场。

[例 1.4.6] 电报方程为

$$\frac{\partial^2 u}{\partial t^2} + a\frac{\partial u}{\partial t} + bu = \frac{\partial^2 u}{\partial x^2}, \tag{1.4.7}$$

其中：a 和 b 是常数。这个方程是由研究电信号在电缆中传输的规律而得出的。电流 I 和电压 U 都满足此式。同样，这个方程还适用于研究脉动血液在动脉中的压力波的传播。

［例 1.4.7］ Klein-Gordon (KG)方程是

$$\Theta\psi + \left(\frac{mc^2}{\hbar}\right)\psi = 0, \tag{1.4.8}$$

$$\Theta \equiv \frac{\partial^2}{\partial t^2} - c^2 \nabla^2, \tag{1.4.9}$$

其中：Θ 是 d'Alembertain 算子；$\hbar$ 是普郎克常数；m 是一物体的静止质量。Klein(1927)和Gordon(1926)推出了在电磁场中描述带电粒子的相对论方程。这是一个守恒的耗散形式，在研究基本粒子时起着十分重要的作用。此方程还可以用来描述宏观上的耗散波。

［例 1.4.8］ 在量子力学中，与时间无关的 Schrödinger 方程为

$$\left(\frac{\hbar^2}{2m}\right)\nabla^2\psi + (E - V)\psi = 0, \tag{1.4.10}$$

其中：$\hbar$ 是普朗克常数；m 是粒子的质量，它的波动函数为 $\psi(x,y,z,t)$；E 是常数；V 是源。如果 $V=0$ 就退化成了 Helmholtz 方程(1.4.6)。

［例 1.4.9］ 线性的 Korteweg-de Vries (KdV)方程是

$$\frac{\partial u}{\partial t} + \alpha\frac{\partial u}{\partial x} + \beta\frac{\partial^3 u}{\partial x^3} = 0, \tag{1.4.11}$$

其中：α,β 是常数。此方程描述了线性的长水波以及等离子波在耗散媒质中的传播。

［例 1.4.10］ 线性 Boussinesq 方程为

$$\frac{\partial^2 u}{\partial t^2} - \alpha^2\nabla^2 u - \beta^2\nabla^2\frac{\partial^2 u}{\partial t^2} = 0, \tag{1.4.12}$$

其中：α,β 是常数。此方程在弹性力学中用以描述杆的纵向波，长的水波以及等离子波。

［例 1.4.11］ 下列方程称为双谐波(biharmonic wave)方程：

$$\frac{\partial^2 u}{\partial t^2} + c^2\nabla^4 u = 0, \tag{1.4.13}$$

其中：c 是常数。在弹性力学中，一个薄弹性盘受到了轻微的振动满足此方程。当 u 与 t 无关时，就退化成了式(1.4.14)，称它为双谐振(Biharmonic)方程，

$$\nabla^4 u = 0, \tag{1.4.14}$$

它描述了满足 Airy 应力函数 $u(x,y,z)$ 的弹性媒质中的应力分布的平衡问题。在流体动力学中，在粘性液体中的流函数 $\psi(x,y,z)$ 也满足此方程。

［例 1.4.12］ 在电场 E 中的电磁波动和极化磁场 P 满足如下方程：

$$\epsilon_0\left(\frac{\partial^2 E}{\partial t^2} - c_0{}^2\frac{\partial^2 E}{\partial x^2}\right) + \frac{\partial^2 P}{\partial t^2} = 0, \tag{1.4.15}$$

$$\left(\frac{\partial^2 P}{\partial t^2} + \omega_0{}^2 P\right) - \epsilon_0\omega_p{}^2 + E = 0, \tag{1.4.16}$$

其中：ϵ_0 是空气的介电常数；ω_0 是振荡的自然频率；c_0 是光在真空中的传播速度；ω_p 是等离子频率。

1.5 二阶线性方程的分类

二元二阶线性偏微分方程的通式为

$$A\frac{\partial^2 u}{\partial x^2}+B\frac{\partial^2 u}{\partial x\partial y}+C\frac{\partial^2 u}{\partial y^2}+D\frac{\partial u}{\partial x}+E\frac{\partial u}{\partial y}+Fu=G, \tag{1.5.1}$$

其中：A,B,C,D,E,F 和 G 都是量 x,y 的函数或常数。

对二阶微分方程的分类主要看其是否有可能通过变换，在某一点将方程(1.5.1)转化成标准(standard)式或经典(canonical)形式。假设

$$\xi=\phi(x,y), \tag{1.5.2a}$$

$$\eta=\psi(x,y)。 \tag{1.5.2b}$$

考虑如何把 x,y 转化成 ξ,η。其中，ϕ 和 ψ 是二阶连续可微函数，而且其雅可比函数 $J(x,y)=\frac{\partial\phi}{\partial x}\frac{\partial\psi}{\partial y}-\frac{\partial\phi}{\partial y}\frac{\partial\psi}{\partial x}$ 在定义域内非零。这样 x 和 y 可以由式(1.5.2a)和(1.5.2b)唯一确定，并可以依次导出以下各式：

$$\frac{\partial u}{\partial x}=\frac{\partial u}{\partial\xi}\frac{\partial\xi}{\partial x}+\frac{\partial u}{\partial\eta}\frac{\partial\eta}{\partial x},$$

$$\frac{\partial u}{\partial y}=\frac{\partial u}{\partial\xi}\frac{\partial\xi}{\partial y}+\frac{\partial u}{\partial\eta}\frac{\partial\eta}{\partial y},$$

$$\frac{\partial^2 u}{\partial x^2}=\frac{\partial^2 u}{\partial\xi^2}\left(\frac{\partial\xi}{\partial x}\right)^2+2\frac{\partial^2 u}{\partial\xi\partial\eta}\frac{\partial\xi}{\partial x}\frac{\partial\eta}{\partial x}+\frac{\partial^2 u}{\partial\eta^2}\left(\frac{\partial\eta}{\partial x}\right)^2+\frac{\partial u}{\partial\xi}\frac{\partial^2\xi}{\partial x^2}+\frac{\partial u}{\partial\eta}\frac{\partial^2\eta}{\partial x^2},$$

$$\frac{\partial^2 u}{\partial y^2}=\frac{\partial^2 u}{\partial\xi^2}\left(\frac{\partial\xi}{\partial y}\right)^2+2\frac{\partial^2 u}{\partial\xi\partial\eta}\frac{\partial\xi}{\partial y}\frac{\partial\eta}{\partial y}+\frac{\partial^2 u}{\partial\eta^2}\left(\frac{\partial\eta}{\partial y}\right)^2+\frac{\partial u}{\partial\xi}\frac{\partial^2\xi}{\partial y^2}+\frac{\partial u}{\partial\eta}\frac{\partial^2\eta}{\partial y^2},$$

$$\frac{\partial^2 u}{\partial x\partial y}=\frac{\partial^2 u}{\partial\xi^2}\frac{\partial\xi}{\partial x}\frac{\partial\xi}{\partial y}+\frac{\partial^2 u}{\partial\xi\partial\eta}\left(\frac{\partial\xi}{\partial x}\frac{\partial\eta}{\partial y}+\frac{\partial\xi}{\partial y}\frac{\partial\eta}{\partial x}\right)+\frac{\partial^2 u}{\partial\eta^2}\frac{\partial\eta}{\partial x}\frac{\partial\eta}{\partial y}+\frac{\partial u}{\partial\xi}\frac{\partial^2\xi}{\partial x\partial y}+\frac{\partial u}{\partial\eta}\frac{\partial^2\eta}{\partial x\partial y}。$$

用以上各式代入式(1.5.1)，可以得到

$$A^*\frac{\partial^2 u}{\partial\xi^2}+B^*\frac{\partial^2 u}{\partial\xi\partial\eta}+C^*\frac{\partial^2 u}{\partial\eta^2}+D^*\frac{\partial u}{\partial\xi}+E^*\frac{\partial u}{\partial\eta}+F^*u=G, \tag{1.5.3}$$

其中：

$$A^*=A\left(\frac{\partial\xi}{\partial x}\right)^2+B\frac{\partial\xi}{\partial x}\frac{\partial\xi}{\partial y}+C\left(\frac{\partial\xi}{\partial y}\right)^2,$$

$$B^*=2A\frac{\partial\xi}{\partial x}\frac{\partial\eta}{\partial x}+B\left(\frac{\partial\xi}{\partial x}\frac{\partial\eta}{\partial y}+\frac{\partial\xi}{\partial y}\frac{\partial\eta}{\partial x}\right)+2C\frac{\partial\xi}{\partial y}\frac{\partial\eta}{\partial y},$$

$$C^*=A\left(\frac{\partial\eta}{\partial x}\right)^2+B\frac{\partial\eta}{\partial x}\frac{\partial\eta}{\partial y}+C\left(\frac{\partial\eta}{\partial y}\right)^2,$$

$$D^*=A\frac{\partial^2\xi}{\partial x^2}+B\frac{\partial\xi}{\partial x}\frac{\partial\xi}{\partial y}+C\frac{\partial^2\xi}{\partial y^2}+D\frac{\partial\xi}{\partial x}+E\frac{\partial\xi}{\partial y},$$

$$E^*=A\frac{\partial^2\eta}{\partial x^2}+B\frac{\partial^2\eta}{\partial x\partial y}+C\frac{\partial^2\eta}{\partial y^2}+D\frac{\partial\eta}{\partial x}+E\frac{\partial\eta}{\partial y},$$

$$F^*=F,$$

$$G^*=G。$$

现在问题是如何选择 ξ 和 η 的函数表达式，使式(1.5.3)成为一个简单式。可以使 $A^*=C^*=0$ 且 $B^*\neq0$，即

$$A^*-A\left(\frac{\partial\xi}{\partial x}\right)^2+B\frac{\partial\xi}{\partial x}\frac{\partial\xi}{\partial y}+C\left(\frac{\partial\xi}{\partial y}\right)^2=0, \tag{1.5.4}$$

$$C^*=A\left(\frac{\partial\eta}{\partial x}\right)^2+B\frac{\partial\eta}{\partial x}\frac{\partial\eta}{\partial y}+C\left(\frac{\partial\eta}{\partial y}\right)^2=0, \tag{1.5.5}$$

令 $\zeta=\xi$ 或 η，这两个式子可以进一步合并成

$$A\left(\frac{\partial\zeta/\partial x}{\partial\zeta/\partial y}\right)^2+B\left(\frac{\partial\zeta/\partial x}{\partial\zeta/\partial y}\right)+C=0。\tag{1.5.6}$$

假设曲面 $\phi(x,y)=C_1$，$\psi(x,y)=C_2$，在这两个曲面上

$$\mathrm{d}\xi=\frac{\partial\xi}{\partial x}\mathrm{d}x+\frac{\partial\xi}{\partial y}\mathrm{d}y=0,\tag{1.5.7a}$$

$$\mathrm{d}\eta=\frac{\partial\eta}{\partial x}\mathrm{d}x+\frac{\partial\eta}{\partial y}\mathrm{d}y=0。\tag{1.5.7b}$$

这些曲面的斜率就是

$$\frac{\mathrm{d}y}{\mathrm{d}x}=-\frac{\partial\xi/\partial x}{\partial\xi/\partial y},\tag{1.5.8a}$$

$$\frac{\mathrm{d}y}{\mathrm{d}x}=-\frac{\partial\eta/\partial x}{\partial\eta/\partial y}。\tag{1.5.8b}$$

因此，这两个曲面的斜率都是方程(1.5.6)的根，即

$$A\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)^2-B\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)+C=0。\tag{1.5.9}$$

而这个方程的根可以表示为

$$\frac{\mathrm{d}y}{\mathrm{d}x}=\frac{1}{2A}(B\pm\sqrt{B^2-4AC})。\tag{1.5.10a,b}$$

这就是方程(1.5.1)的特征方程(characteristic equations)。它们的解称为特征曲线(characteristic curves)，也就是方程的特征根(characteristics)。常微分方程(1.5.10a,b)的解定义了两个曲线族 $\phi(x,y)=C_1$，$\psi(x,y)=C_2$，这里有三种情况需要讨论。

情况Ⅰ $B^2-4AC>0$

如果 $B^2-4AC>0$，则称之为双曲型(hyperbolic)方程。对式(1.5.10a,b)分别沿着 $\phi(x,y)=C_1$，$\psi(x,y)=C_2$ 积分。因为 $A^*=C^*=0$ 且 $B^*\neq0$，两边同时除以 B^*，式(1.5.3)就可以简化成

$$\frac{\partial^2u}{\partial\xi\partial\eta}=-\frac{1}{B^*}\left(D^*\frac{\partial u}{\partial\xi}+E^*\frac{\partial u}{\partial\eta}+F^*u-G^*\right)=H_1。\tag{1.5.11}$$

这就是双曲型方程的第一标准形式(first canonical form of the hyperbolic equation)。

如果引进新的变量

$$\alpha=\xi+\eta,\tag{1.5.12a}$$

$$\beta=\xi-\eta,\tag{1.5.12b}$$

那么

$$\frac{\partial u}{\partial\xi}=\frac{\partial u}{\partial\alpha}\frac{\partial\alpha}{\partial\xi}+\frac{\partial u}{\partial\beta}\frac{\partial\beta}{\partial\xi}=\frac{\partial u}{\partial\alpha}+\frac{\partial u}{\partial\beta},$$

$$\frac{\partial u}{\partial\eta}=\frac{\partial u}{\partial\alpha}\frac{\partial\alpha}{\partial\eta}+\frac{\partial u}{\partial\beta}\frac{\partial\beta}{\partial\eta}=\frac{\partial u}{\partial\alpha}-\frac{\partial u}{\partial\beta},$$

$$\frac{\partial\left(\frac{\partial u_\eta}{\partial\eta}\right)}{\partial\xi}=\frac{\partial\left(\frac{\partial u_\eta}{\partial\eta}\right)}{\partial\alpha}\frac{\partial\alpha}{\partial\xi}+\frac{\partial\left(\frac{\partial u_\eta}{\partial\eta}\right)}{\partial\beta}\frac{\partial\beta}{\partial\xi}$$

$$=\frac{\partial\left(\frac{\partial u}{\partial\alpha}-\frac{\partial u}{\partial\beta}\right)}{\partial\alpha}\cdot1+\frac{\partial\left(\frac{\partial u}{\partial\alpha}-\frac{\partial u}{\partial\beta}\right)}{\partial\beta}\cdot1$$

$$=\frac{\partial^2 u}{\partial \alpha^2}-\frac{\partial^2 u}{\partial \beta^2},$$

式(1.5.11)就成为

$$\frac{\partial^2 u}{\partial \alpha^2}-\frac{\partial^2 u}{\partial \beta^2}=H_2\left(\alpha,\beta,u,\frac{\partial u}{\partial \alpha},\frac{\partial u}{\partial \beta}\right)。\tag{1.5.13}$$

这就是双曲型方程的第二标准形式(second canonical form of the hyperbolic equation)。这里还要强调一下，特征线在双曲型方程的研究中起着非常重要的作用。

情况Ⅱ　$B^2-4AC=0$

这里只有一个特征线族，因为在式(1.5.10a,b)中

$$\frac{\mathrm{d}y}{\mathrm{d}x}=\frac{B}{2A}。\tag{1.5.14}$$

对此函数沿着 $\xi=\phi(x,y)=\text{const.}$（或者 $\eta=\psi(x,y)=\text{const.}$）积分，因为 $B^2=4AC$ 且 $A^*=0$，所以可以得到

$$0=A^*=A\left(\frac{\partial \xi}{\partial x}\right)^2+B\frac{\partial \xi}{\partial x}\frac{\partial \xi}{\partial y}+C\left(\frac{\partial \xi}{\partial y}\right)^2=\left(\sqrt{A}\frac{\partial \xi}{\partial x}+\sqrt{C}\frac{\partial \xi}{\partial y}\right)^2。$$

可以推出

$$\begin{aligned}B^*&=2A\frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial x}+B\left(\frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial y}+\frac{\partial \xi}{\partial y}\frac{\partial \eta}{\partial x}\right)+2C\frac{\partial \xi}{\partial y}\frac{\partial \eta}{\partial y}\\&=2\left(\sqrt{A}\frac{\partial \xi}{\partial x}+\sqrt{C}\frac{\partial \xi}{\partial y}\right)\left(\sqrt{A}\frac{\partial \eta}{\partial x}+\sqrt{C}\frac{\partial \eta}{\partial y}\right)\\&=0。\end{aligned}$$

对于任何一个给定的与 ξ 无关的 η，例如 $\eta=y$，那么在定义域内的雅可比函数处处非零。

用 $C^*\neq 0$ 除以式(1.5.3)得到

$$\frac{\partial^2 u}{\partial \eta^2}=H_3\left(\xi,\eta,u,\frac{\partial u}{\partial \xi},\frac{\partial u}{\partial \eta}\right)。\tag{1.5.15}$$

这就是标准的抛物型方程形式(canonical form of the parabolic equation)。

如果选择($\eta=\psi(x,y)=\text{const.}$)作为式(1.5.14)的积分路径，那么式(1.5.3)就可以化成

$$\frac{\partial^2 u}{\partial \xi^2}=H_3^*\left(\xi,\eta,u,\frac{\partial u}{\partial \xi},\frac{\partial u}{\partial \eta}\right)。\tag{1.5.16}$$

称满足 $B^2-4AC=0$ 的方程为抛物型(parabolic)方程。

情况Ⅲ　$B^2-4AC<0$

如果 $B^2-4AC<0$，称椭圆型方程。在这种情况下，方程(1.5.10a,b)没有实数解，仅有两个复数的特征线族，且根 ξ,η 是一对共轭的复数根。引进新的变量

$$\alpha=\frac{1}{2}(\xi+\eta),\tag{1.5.17a}$$

$$\beta=\frac{1}{2\mathrm{i}}(\xi-\eta),\tag{1.5.17b}$$

所以 $\xi=\alpha+\beta\mathrm{i},\eta=\alpha-\beta\mathrm{i}$。用式(1.5.17a,b)来把式(1.5.3)化简成

$$A^{**}\frac{\partial^2 u}{\partial \alpha^2}+B^{**}\frac{\partial^2 u}{\partial \alpha\partial \beta}+C^{**}\frac{\partial^2 u}{\partial \beta^2}=H_4\left(\alpha,\beta,u,\frac{\partial u}{\partial \alpha},\frac{\partial u}{\partial \beta}\right),\tag{1.5.18}$$

此方程的系数和方程(1.5.3)的系数相同。容易证明，当且仅当 $A^{**}=C^{**}$，且 $B^{**}=0$ 时，只

有 $A^* = 0, C^* = 0$ 才满足

$$A^{**} - C^{**} \pm \mathrm{i}B^{**} = 0,$$

所以，由式(1.5.18)两边同时除以 A^{**} 可得

$$\frac{\partial^2 u}{\partial \alpha^2} + \frac{\partial^2 u}{\partial \beta^2} = \frac{H_4}{A^{**}} = H_5\left(\alpha, \beta, u, \frac{\partial u}{\partial \alpha}, \frac{\partial u}{\partial \beta}\right)。 \tag{1.5.19}$$

这就是椭圆型方程的标准形式(canonical form of the elliptic equation)。

应该指出，在两个独立变量的情况下，总可以找到一种变换，把方程变换成标准形式。但是，在多变量的情况下，找到这种变换是很困难的，可以说大多情况下是不存在的。

在数学和物理的许多领域经常会出现这三种偏微分方程。一般来说，对于椭圆型的方程常研究边值问题的解，而对于抛物型方程和双曲型方程常讨论的是初值问题。

1.6 非线性偏微分方程

两个自变量 x, y 的一阶非线性偏微分方程可以写成

$$F\left(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) = 0。 \tag{1.6.1}$$

同样，二阶非线性偏微分方程可以写成

$$F\left(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}, \frac{\partial^2 u}{\partial x \partial y}\right) = 0。 \tag{1.6.2}$$

以此类推，可得高阶非线性偏微分方程的一般形式。

同样，可以将偏微分方程写成算子形式

$$L_x u(\boldsymbol{x}) = f(\boldsymbol{x}), \tag{1.6.3}$$

式中：L_x 是偏微分算子；而 $f(\boldsymbol{x})$ 是两个或两个以上自变量 $\boldsymbol{x} = (x, y, \cdots)$ 的函数。如果 L_x 不是线性算子，方程(1.6.3)是非线性偏微分方程。如果 $f(\boldsymbol{x}) \neq 0$，方程(1.6.3)是非齐次非线性偏微分方程。否则，如果 $f(\boldsymbol{x}) = 0$，方程(1.6.3)是齐次非线性偏微分方程。

如果在满足某些收敛条件的情况下，线性叠加原理可以用于解线性偏微分方程。这个原理通常对给定解进行组合以得到一个新的解。但对于非线性偏微分方程，线性叠加原理一般不再产生新解。由于解线性偏微分方程的大部分解析方法不再适用于解非线性偏微分方程，因此，非线性偏微分方程通常采用数值计算方法解。与线性偏微分方程一样，非线性偏微分方程的解也有存在性、唯一性和稳定性问题，但这些问题的解答要比线性偏微分方程困难得多。

1.7 经典非线性偏微分方程

[例 1.7.1] 最简单的一阶非线性波动方程

$$\frac{\partial u}{\partial t} + c(u)\frac{\partial u}{\partial x} = 0, \qquad x \in \mathbf{R}, t > 0, \tag{1.7.1}$$

其中：$c(u)$ 是 u 的给定函数，可用于描述高速公路的车流量，空气动力学中的激波，水力学中的潮涌波等。

[例 1.7.2] 非线性 Klein-Gordon 方程为

$$\frac{\partial^2 u}{\partial t^2} - c^2\Delta^2 u + V'(u) = 0, \tag{1.7.2}$$

其中：c 是常数；$V'(u)$是 u 的非线性函数。该方程在很多物理问题中出现，如非线性散射和非线性介子理论。

［例 1.7.3］ Burgers 方程为

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \qquad x \in \mathbf{R},\ t > 0, \tag{1.7.3}$$

其中：ν 是运动粘性系数。这是流体力学中处理耗散波的最简单非线性方程。最早由 Burgers 提出以描述一维涡流，也用于描述在粘性介质中的声波，充满粘性流体的弹性管中的波等。

［例 1.7.4］ Fisher 方程为

$$\frac{\partial u}{\partial t} - v\frac{\partial^2 u}{\partial x^2} = k\left(u - \frac{u}{\kappa}\right), \qquad x \in \mathbf{R},\ t > 0, \tag{1.7.4}$$

其中：v,k 和 κ 是常数。该方程在很多生物和化学大系统中用以研究扩散的非线性模型。

由于非线性偏微分方程种类繁多，详细讨论已超出本书范围。这里只能列举一些最基本的方程，感兴趣的读者可以参阅文献[1]～[3]。

1.8　变分原理和 Euler-Lagrange 方程

经典的 Euler-Lagrange 变分问题是，在域 $D \subset \mathbf{R}^2$ 中寻找函数 $u(x,y)$，满足在 D 的边界 ∂D上的边界条件，并使泛函

$$I[u(x,y)] = \iint_D F\left(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right)\mathrm{d}x\mathrm{d}y \tag{1.8.1}$$

达到极值。式中函数 F 的定义域为 D，且具有二阶连续偏导数。

根据一个自变量函数的泛函的性质，函数 I 的一阶变分

$$\delta I(u,h) = I(u+h) - I(u)。 \tag{1.8.2}$$

用 Taylor 级数展开，可得

$$\delta I = \iint_D \left(v\frac{\partial F}{\partial u} + \frac{\partial v}{\partial x}\frac{\partial F}{\partial p} + \frac{\partial v}{\partial y}\frac{\partial F}{\partial q}\right)\mathrm{d}x\mathrm{d}y, \tag{1.8.3}$$

其中：$v = v(x,y)$ 且在边界∂D 上为零，而 $p = \frac{\partial u}{\partial x}$和 $q = \frac{\partial u}{\partial y}$。

函数 I 取得其极值的一个必要条件是，它的一阶变分为零，即

$$\delta I = \iint_D \left(v\frac{\partial F}{\partial u} + \frac{\partial v}{\partial x}\frac{\partial F}{\partial p} + \frac{\partial v}{\partial y}\frac{\partial F}{\partial q}\right)\mathrm{d}x\mathrm{d}y = 0,$$

可写成

$$\begin{aligned}\delta I &= \iint_D v\left(\frac{\partial F}{\partial u} - \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial p}\right) - \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial q}\right)\right)\mathrm{d}x\mathrm{d}y \\ &\quad + \iint_D \left(v\left(\frac{\partial}{\partial x}\left(\frac{\partial F}{\partial p}\right) + \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial q}\right)\right) + \left(\frac{\partial v}{\partial x}\frac{\partial F}{\partial p} + \frac{\partial v}{\partial y}\frac{\partial F}{\partial q}\right)\right)\mathrm{d}x\mathrm{d}y \\ &= 0。\end{aligned} \tag{1.8.4}$$

假设在边界∂D 上有分段连续的斜率，根据 Green 定理，式(1.8.4)可简化为

$$\delta I = \iint_D v\left[\frac{\partial F}{\partial u} - \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial p}\right) - \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial q}\right)\right]\mathrm{d}x\mathrm{d}y + \int v\left(\frac{\partial F}{\partial u}\mathrm{d}x - \frac{\partial F}{\partial q}\mathrm{d}y\right) = 0。 \quad (1.8.5)$$

因为在边界∂D上函数$v=0$，方程(1.8.5)中第二个积分为零。而函数$v=v(x,y)$在域D内不为零，要使方程(1.8.5)为零的必要条件是

$$\frac{\partial F}{\partial u} - \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial p}\right) - \frac{\partial}{\partial y}\left(\frac{\partial F}{\partial q}\right) = 0, \quad (1.8.6)$$

即函数$u(x,y)$使泛函(1.8.1)取极值的必要条件是满足偏微分方程(1.8.6)。这就是具有两个独立自变量的 Euler-Lagrange 方程。仿此，也可以得到两个以上独立自变量的 Euler-Lagrange 方程。

特别是，如果函数$u=u(x)$只有一个独立自变量x，那么泛函式(1.8.1)为

$$I[u(x)] = \int_a^b F(x,u,u')\mathrm{d}x, \quad (1.8.7)$$

其中：$u'=\frac{\mathrm{d}u}{\mathrm{d}x}$，并且 Euler-Lagrange 方程(1.8.6)简化成标准形式

$$\frac{\partial F}{\partial u} - \frac{\mathrm{d}}{\mathrm{d}x}\left[\frac{\partial F}{\partial u'}\right] = 0。 \quad (1.8.8)$$

1.9 极值原理

极值原理是偏微分方程理论中一个基本且重要的结论。简要说，对于某些类型的偏微分方程解的最大(小)值在解定义域的空间或时间边界上。这个原理有助于证明解的唯一性和其他一些偏微分方程的基本性质。例如：某个偏微分方程描述两条曲线的变形，而且第一条曲线严格在第二条曲线之内，那么这两条曲线间的距离应满足极值原理。极值原理也可用于检验变形曲线的自交叉点随时间增加或减少。有关偏微分方程的极值原理，可参阅文献[3]。

[例 1.9.1] 假设连续函数$u(x):[a,b]\to\Re$在C点上取得极值，其二阶偏导数$\frac{\partial^2 u}{\partial x^2}$是连续的，且在点$c\in[a,b]$取得局部最大，根据数学分析的基本定理有

$$\left.\frac{\partial u}{\partial x}\right|_{x=c} = 0, \quad \left.\frac{\partial^2 u}{\partial x^2}\right|_{x=c} \leqslant 0。 \quad (1.9.1)$$

假设，在开区间(a,b)，u满足

$$\frac{\partial^2 u}{\partial x^2} + g(x)\frac{\partial u}{\partial x} > 0,$$

其中：$g(x)$是有界函数。很明显，关系式(1.9.1)在开区间(a,b)无法满足，极值点只可能在端点a或b。这是极值原理的一个例子。

[例 1.9.2] 假设$u(x,y)$是二次可微的，且定义拉普拉斯算子$\Delta=\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}$，拉普拉斯方程为

$$\Delta u = 0 。$$

在域D中，满足上述方程的函数u称为调和函数。如果u在域D中取得最大，那么u的一阶导数为零，二阶导数为负(或等于零)。因此在极值点，有

$$\Delta u \leqslant 0,$$

但是,如果在域 D 中,u 处处满足

$$\Delta u > 0,$$

则 u 不能取得其极值。因此有以下定理:

定理 1.9.1　设在域 D 中有 $\Delta u \geqslant 0$,如果 u 在域 D 任意点取得极值,则 u 在域 D 中是常数。

上述定理只适用于具有有界函数的统一椭圆算子。下面举个例子说明这个简单的结果对于证明解的唯一性是很有用的。给定两个函数 $f(x,y)$ 和 $g(s)$,寻找在有界区间 D 中二次可微,在 $D \cup \partial D$ 中连续的函数 $u(x,y)$,并在域 D 中满足

$$\Delta u = f(x,y)。$$

在边界 ∂D 上为

$$u = g(x)。$$

假设 u_1 和 u_2 是方程的解,定义 $u=u_1-u_2$,容易证明

$$\Delta u = 0。$$

由于 u 在 $D \cup \partial D$ 上连续且在 ∂D 上 $u=0$,因此 $u \leqslant 0$。根据极值原理,u 在域 D 中不可能达到极值。同理,对于 $-u$,有 $u \geqslant 0$,因此在域 D 中 $u=0$,且 $u_1=u_2$。

本书中抛物线方程用得较多,因此考虑抛物线方程

$$\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial t} = f(x,t),$$

这是热传导方程。假设在平面 (x,y) 的域 $E=\{0<x<a, 0<t\leqslant\tau\}$ 中,u 满足

$$\frac{\partial^2 u}{\partial x^2} - \frac{\partial u}{\partial t} > 0,$$

那么,u 的极值只可能在边界 $S_1=\{x=0, 0\leqslant t\leqslant\tau\}$,$S_2=\{0\leqslant x\leqslant a, t=0\}$ 和 $S_3=\{x=a, 0\leqslant t\leqslant\tau\}$ 中的一条边界上。

参　考　文　献

[1] Debnath L. Nonlinear partial differential equations for scientists and engineers. Boston: Birkhauser, 1997

[2] Isaacson E, Keller H B. Analysis of Numerical Methods. New York: Dover Publication, Inc., 1994

[3] Protter M H, Weinberger H. Maximum Principles in Differential Equations, New York: Springer-Verlag, 1984

[4] Aubert G, Kornprobst P. Mathematical problem in image processing-Partial differential equation and the calculus of variations. New York: Springer Verlag, 2001

[5] Sapiro G. Geometric partial differential equations and image analysis. Cambridge: Cambridge University Press, 2001

[6] 刘式适,刘式达. 物理学中的非线性方程. 北京:北京大学出版社,2000

[7] 刘盾. 实用数学物理方程. 重庆:重庆大学出版社,1999

[8] 潘祖梁,陈仲慈. 工程技术中的偏微分方程. 杭州:浙江大学出版社,2000

[9] 谢鸿政,杨枫林. 数学物理方程. 北京:科学出版社,2001

[10] 徐建平,桂子鹏. 变分方法. 上海:同济大学出版社,1999

第 2 章　有限差分法

通常，图像中内容比较复杂，不易用简单的数学模型描述。因此，用偏微分方程处理图像问题，很难得到解析解，一般都求其数值逼近（或近似）解。常用的偏微分方程数值解的方法很多，如：有限差分法，有限单元法，边界元法等。但在图像偏微分方程中，用得最多的是有限差分法。

本章从有限差分方程解的基本概念开始，讨论它与原偏微分方程解的关系。使用有限差分法时应当了解算法的收敛性和在何种条件下达到收敛以及解的准确性。对于差分方程的收敛性，最一般的方法是从一致性、稳定性和 Lax 定理来讨论的。Lax 定理给出了一致性和稳定性与差分方程收敛性之间的关系。

2.1　有限差分法

考虑一个初始边界值问题的偏微分方程

$$\frac{\partial v}{\partial t}=\mu\frac{\partial^2 v}{\partial x^2},\quad x\in(0,1),t>0; \tag{2.1.1}$$

$$v(x,0)=f(x),\quad x\in[0,1]; \tag{2.1.2}$$

$$v(0,t)=a(t),\quad v(1,t)=b(t),\quad t\geqslant 0。 \tag{2.1.3}$$

式中：$f(0)=a(0)$，且 $f(1)=b(0)$。

这个方程的解析解是容易求的。为了介绍有限差分法，将这个连续偏微分方程转化成离散代数方程。即，先将空间域分成均匀的网格点（或节点）$\Delta x=1/M$，如图 2.1 所示。

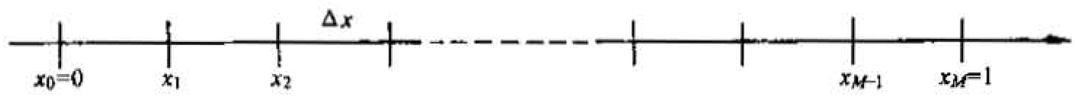


图 2.1　间隔[0,1]分成均匀的网格点

对于某个网格，其中网格点记为点 x_k，$k=0,\cdots,M$，这里 $x_k=k\Delta x$。同样，可以在时间轴上分成均匀的网格点 Δt。

数值解就是求这些网格点上解的近似值或逼近值。定义 u_k^n 为点（$k\Delta x,n\Delta t$）或网格点（k，n）上的函数。函数 u_k^n 为方程（2.1.1）～（2.1.3）在点（$k\Delta x,n\Delta t$）解的逼近值。

下一步就是在这些网格上导出方程（2.1.1）～（2.1.3）的逼近表达式。注意到

$$\frac{\partial v}{\partial t}(x,t)=\lim_{\Delta t\to 0}\frac{v(x,t+\Delta t)-v(x,t)}{\Delta t},$$

因此，可以将点（$k\Delta x,n\Delta t$）上的$\frac{\partial v}{\partial t}$近似为

$$\frac{u_k^{n+1}-u_k^n}{\Delta t}。 \tag{2.1.4}$$

同样，可以将点$(k\Delta x, n\Delta t)$上的$\frac{\partial^2 v}{\partial x^2}$近似为

$$\frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\Delta x^2}。\tag{2.1.5}$$

这样，在点$(k\Delta x, n\Delta t)$上偏微分方程(2.1.1)可利用式(2.1.4)和(2.1.5)近似为

$$\frac{u_k^{n+1} - u_k^n}{\Delta t} = \mu \frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\Delta x^2},\tag{2.1.6}$$

或

$$u_k^{n+1} = u_k^n + \mu \frac{\Delta t}{\Delta x^2}(u_{k+1}^n - 2u_k^n + u_{k-1}^n)。\tag{2.1.7}$$

最后，初始条件和边界条件可以近似为

$$u_k^0 = f(k\Delta x), \quad k = 0, \cdots, M,\tag{2.1.8}$$

$$u_0^{n+1} = a[(n+1)\Delta t], \quad n = 0, \cdots,\tag{2.1.9}$$

和

$$u_M^{n+1} = b[(n+1)\Delta t], \quad n = 0, \cdots,\tag{2.1.10}$$

适当选择Δx(或M)和Δt，然后解离散化的代数方程(2.1.7)～(2.1.10)，就可以得到原方程(2.1.1)～(2.1.3)的逼近解。

解代数方程(2.1.7)～(2.1.10)能直接得到第$(n+1)$时间的变量，因此称之为显式格式。

2.2 截断误差

上节推导逼近偏微分方程(2.1.1)的有限差分方程时，用了近似式

$$\frac{\partial v}{\partial t}(n\Delta t, k\Delta x) \approx \frac{u_k^{n+1} - u_k^n}{\Delta t}。$$

根据泰勒级数展开公式，有

$$v_k^{n+1} = v[k\Delta x, (n+1)\Delta t] = v(k\Delta x, n\Delta t) + \frac{\partial v}{\partial t}(k\Delta x, n\Delta t)\frac{\Delta t}{1!} + \frac{\partial^2 v}{\partial t^2}(k\Delta x, n\Delta t)\frac{\Delta t^2}{2!} + \cdots,\tag{2.2.1}$$

因此，有

$$\frac{v_k^{n+1} - v_k^n}{\Delta t} = \frac{\partial v}{\partial t}(k\Delta x, n\Delta t) + \frac{\Delta t}{2}\left(\frac{\partial^2 v}{\partial t^2}\right)_k^n + \cdots。\tag{2.2.2}$$

假设，在$(k\Delta x, n\Delta t)$上，v的高阶导数是有界的，则记

$$\frac{v_k^{n+1} - v_k^n}{\Delta t} = \frac{\partial v}{\partial t}(k\Delta x, n\Delta t) + O(\Delta t),\tag{2.2.3}$$

定义截断误差

$$\tau_n^k = \frac{v_k^{n+1} - v_k^n}{\Delta t} - \frac{\partial v}{\partial t}(k\Delta x, n\Delta t) = O(\Delta t)。$$

在大多数情况下，由于截断误差与Δt同阶，如果Δt充分小，则$\frac{u_k^{n+1} - u_k^n}{\Delta t}$能很好地逼近$\frac{\partial v}{\partial t}$。但在极少数情况下，方程的解可能随时间剧烈变化，造成较大的截断误差。

同样可得单边差分格式：

$$\frac{v_{k+1}^n - v_k^n}{\Delta x} = \frac{\partial v}{\partial x}(k\Delta x, n\Delta t) + O(\Delta x), \tag{2.2.4}$$

$$\frac{v_k^n - v_{k-1}^n}{\Delta x} = \frac{\partial v}{\partial x}(k\Delta x, n\Delta t) + O(\Delta x) \tag{2.2.5}$$

和中心差分格式：

$$\frac{v_{k+1}^n - v_{k-1}^n}{2\Delta x} = \frac{\partial v}{\partial x}(k\Delta x, n\Delta t) + O(\Delta x^2)。 \tag{2.2.6}$$

可以看出，中心差分逼近的截断误差与 Δx^2 同阶，而单边差分与 Δx 同阶，因此中心差分对于一阶导数的逼近要比单边差分准确。

对于偏微分方程(2.1.1)的差分逼近为

$$\begin{aligned}&\frac{\partial v}{\partial t}(k\Delta x, n\Delta t) - \mu\frac{\partial^2 v}{\partial x^2}(k\Delta x, n\Delta t)\\ =&\frac{v_k^{n+1} - v_k^n}{\Delta t} - \frac{\mu}{\Delta x^2}(v_{k+1}^n - 2v_k^n + v_{k-1}^n) + O(\Delta t) + O(\Delta x^2)。\end{aligned} \tag{2.2.7}$$

可见，差分方程(2.1.6)逼近偏微分方程(2.1.1)的准确度关于 Δt 是一阶的，而关于 Δx 是二阶的。换句话说，如果 $v=v(x,t)$ 是偏微分方程的解，那么差分方程的截断误差，关于 Δt 是一阶的，而关于 Δx 是二阶的。

如果用 $\frac{v_{k+1}^n - v_{k-1}^n}{2\Delta x}$ 逼近 $\frac{\partial v}{\partial t}$，则式(2.1.7)成为

$$u_k^{n+1} = u_k^{n-1} + 2\mu\frac{\Delta t}{\Delta x^2}(u_{k+1}^n - 2u_k^n + u_{k-1}^n)。 \tag{2.2.8}$$

这样，差分格式(2.2.8)在空间和时间上截断误差都是二阶的。这种格式称为三层格式，或 leapfrog 格式。

2.3 选择 Δx 和 Δt

除了采用合适的差分方程格式可以减少截断误差外，适当地选用 Δx 和 Δt 也是减少截断误差的一个途径。迭代格式(2.1.7)，如果适当选择 Δx 和 Δt 可以提高精度的阶次。从展开式(2.2.7)，有

$$\begin{aligned}\left(\frac{\partial v}{\partial t} - \mu\frac{\partial^2 v}{\partial x^2}\right)\Bigg|_k^n - \left[\frac{\delta_{t+}}{\Delta t} - \frac{\mu}{\Delta x^2}\delta^2\right]v_k^n =& -\frac{\Delta t}{2}\left(\frac{\partial^2 v}{\partial t^2}\right)_k^n - \frac{\Delta t^2}{3!}\left(\frac{\partial^3 v}{\partial t^3}\right)_k^n - \cdots\\ &+\mu\left[2\frac{\Delta x^2}{4!}\left(\frac{\partial^4 v}{\partial x^4}\right)_k^n + 2\frac{\Delta x^4}{6!}\left(\frac{\partial^6 v}{\partial x^6}\right)_k^n + \cdots\right],\end{aligned} \tag{2.3.1}$$

因此

$$\frac{\partial v}{\partial t} = \mu\frac{\partial^2 v}{\partial x^2},$$

$$\frac{\partial^2 v}{\partial t^2} = \mu\frac{\partial^3 v}{\partial t\partial x^2} = \mu\frac{\partial^2\left(\frac{\partial v}{\partial t}\right)}{\partial t^2} = \mu\frac{\partial^2\left(\mu\frac{\partial^2 v}{\partial x^2}\right)}{\partial x^2} = \mu^2\frac{\partial^4 v}{\partial x^4}。$$

将上式代入式(2.3.1)有

$$\left(\frac{\partial v}{\partial t}-\mu\frac{\partial^2 v}{\partial x^2}\right)_k^n-\left[\frac{\delta_{t+}}{\Delta t}-\frac{\mu}{\Delta x^2}\delta^2\right]v_k^n=-\frac{\Delta t}{2}\mu^2\left(\frac{\partial^4 v}{\partial x^4}\right)_k^n-\frac{\Delta t^2}{3!}\left(\frac{\partial^3 v}{\partial t^3}\right)_k^n-\cdots$$

$$+\mu\left[2\frac{\Delta x^2}{4!}\left(\frac{\partial^4 v}{\partial x^4}\right)_k^n+2\frac{\Delta x^4}{6!}\left(\frac{\partial^6 v}{\partial x^6}\right)_k^n+\cdots\right]$$

$$=\mu\left(-\mu\frac{\Delta t}{2}+2\frac{\Delta x^2}{4!}\right)\left(\frac{\partial v^4}{\partial x^4}\right)_k^n+O(\Delta t^2)+O(\Delta x^4)。$$

因此，如果选择 $\Delta t=\dfrac{\Delta x^2}{6\mu}$，那么该方程在时间上是二阶的，而在空间上是四阶的。

2.4　非齐次方程和边界条件

本节讨论与方程(2.1.1)相似的一类非齐次偏微分方程

$$\frac{\partial v}{\partial t}=\mu\frac{\partial^2 v}{\partial x^2}+F(x,t),\quad x\in(0,1),t>0。\tag{2.4.1}$$

只要将式(2.2.8)稍加修改就可以解非齐次偏微分方程(2.4.1)。记 $F_k^n=F(k\Delta x,n\Delta t)$，有差分格式

$$u_k^{n+1}=u_k^n+\frac{\mu\Delta t}{\Delta x^2}\delta^2 u_k^n+\Delta tF_k^n。\tag{2.4.2}$$

如果边界条件与时间有关，在随时间变化的差分格式中也应考虑边界条件。

2.5　收敛性

2.5.1　初值问题

精心选择有限差分格式可以得到相应偏微分方程的近似解。实际离散化过程中，要确保差分方程的解能以任意精度逼近其偏微分方程的精确解。需要证明差分方程解是否能收敛到偏微分方程的解。差分方程收敛性不是容易解决的问题。以初值问题为例，考虑偏微分方程：

$$Lv=F,\tag{2.5.1}$$

其中：函数 v 和 F 的定义域是整个实轴；初始条件为 $v(x,0)=f(x)$。假定已经求得此问题在离散网格(采样间隔为 Δx 和 Δt)上的一个近似解 u_k^n(在差分格式中用 L_k^n 表示，上标 n 表示时间步骤，下标 k 表示空间网格点)，并且 u_k^n 满足初始条件：$u_k^0=f(k\Delta x),k=-\infty,\cdots,\infty$。令 v 为初值问题的精确解，有下述定义。

定义 2.5.1　如果对任意 x 和 t，当 Δx 和 Δt 收敛到 0，并且$[k\Delta x,(n+1)\Delta t]$收敛到$(x,y)$时，$u_k^n$ 收敛到 $v(x,y)$，则称逼近偏微分方程的差分格式 $Lv=F$ 是点(pointwise)收敛的。

[例 2.5.1]　证明差分格式

$$u_k^{n+1}=(1-2r)u_k^n+r(u_{k+1}^n+u_{k-1}^n),\tag{2.5.2}$$

$$u_k^0=f(k\Delta x),\tag{2.5.3}$$

式中：$r=\mu\Delta t/\Delta x^2,0\leqslant r\leqslant 1/2$；点收敛于初始值问题

$$\frac{\partial v}{\partial t}=\mu\frac{\partial^2 v}{\partial x^2},\quad x\in\Re,t>0;\tag{2.5.4}$$

$$v(x,0)=f(x),\quad x\in\Re。\tag{2.5.5}$$

［解］ 由于这个初始值问题的定义域为所有实轴 $\Re$ 上的点，网格点 k 为 $-\infty<k<\infty$ 上的整数。

令初始值问题(2.5.4)和(2.5.5)的精确解为 $v=(x,t)$，并设

$$z_k^n=u_k^n-v(k\Delta x,n\Delta t)。\tag{2.5.6}$$

如果将 v 代入方程(2.2.7)［使方程(2.2.7)的等号左边为零］并两边乘以 Δt，可见 $v_k^n=v(k\Delta t,n\Delta t)$ 满足

$$v_k^{n+1}=(1-2r)v_k^n+r(v_{k+1}^n+v_{k-1}^n)+O(\Delta t^2)+O(\Delta t\Delta x^2)。\tag{2.5.7}$$

然后将式(2.5.2)减去式(2.5.7)，有

$$z_k^{n+1}=(1-2r)z_k^n+r(z_{k+1}^n+z_{k-1}^n)+O(\Delta t^2)+O(\Delta t\Delta x^2)。\tag{2.5.8}$$

如果 $0<r\leqslant 1/2$，式(2.5.8)右边的系数是非负的，并且

$$\begin{aligned}|z_k^{n+1}|&\leqslant(1-2r)|z_k^n|+r|z_{k+1}^n|+r|z_{k-1}^n|+A(\Delta t^2+\Delta t\Delta x^2)\\&\leqslant Z^n+A(\Delta t^2+\Delta t\Delta x^2),\end{aligned}$$

其中：A 是 v 的高阶无穷小量；$Z^n=\sup_k\{|z_k^n|\}$。然后在上式的左边取 k 的上界得到

$$Z^{n+1}\leqslant Z^n+A(\Delta t^2+\Delta t\Delta x^2)。\tag{2.5.9}$$

重复式(2.5.9)得到

$$\begin{aligned}Z^{n+1}&\leqslant Z^n+A(\Delta t^2+\Delta t\Delta x^2)\\&\leqslant Z^{n-1}+2A(\Delta t^2+\Delta t\Delta x^2)\\&\ \ \vdots\\&\leqslant Z^0+(n+1)A(\Delta t^2+\Delta t\Delta x^2)。\end{aligned}$$

因为，$Z^0=0$，

$$|u_k^{n+1}-v[k\Delta x,(n+1)\Delta t]|\leqslant Z^{n+1},\quad \text{且}(n+1)\Delta t\to t,$$

当 $\Delta t,\Delta x\to 0$

$$|u_k^{n+1}-v[k\Delta x,(n+1)\Delta t]|\leqslant(n+1)\Delta tA(\Delta t+\Delta x^2)\to 0。\tag{2.5.10}$$

可知，对于任意的 x 和 t，当 Δt 和 Δx 趋于零时，$[k\Delta x,(n+1)\Delta t]\to(x,t)$，$u_k^n$ 趋于 $v(x,t)$。证明过程中应该注意以下三点：

(1) 式(2.5.10)中 $(n+1)$ 对收敛有较大影响，因此条件 $(n+1)\Delta t\to t$ 是极其重要的。

(2) 式(2.5.10)中条件 $0<r\leqslant 1/2$ 也很重要，这样对时间步长 Δt 有了约束。

(3)证明过程中要求高阶无穷小量 A 在所有网格点 k 和时间 $\Re\times[0,t]$ 都是均匀有界的，所有有界序列 l_∞ 的范数的上界(sup-norm)表示为

$$\|\{a_k\}\|_\infty=\sup_{-\infty<k<\infty}|a_k|。$$

令 $\boldsymbol{u}^n=(\cdots u_{-1}^n,u_0^n,u_1^n,\cdots)^{\mathrm T}$，$\boldsymbol{v}^n=(\cdots v_{-1}^n,v_0^n,v_1^n,\cdots)^{\mathrm T}$。已经证明，对于 t，当 $(n+1)\Delta t$ 逼近 t 时，$\boldsymbol{u}^{n+1}$ 逼近 $v(\cdot,t)$。这里“逼近”的概念是，当 Δx 和 Δt 趋向于 0 时，$\boldsymbol{u}^{n+1}-\boldsymbol{v}^{n+1}$ 的范数上界趋向于 0。

通常，点收敛的证明比较困难，因此很少使用。取而代之的是，用偏微分方程解和差分方程解的差的范数证明收敛。这是上述按范数上界收敛的推广。为方便，以 $\boldsymbol{u}^n$ 表示差分方程解的值 $\{u_k^n\}$ 组成的矢量，以 $\boldsymbol{v}^n$ 表示偏微分方程在网格点处解的值 $v(k\Delta x,n\Delta t)$ 组成的矢量。

定义 2.5.2　在时间 t，如果 $\Delta x\to 0$，$\Delta t\to 0$ 且 $(n+1)\Delta t\to t$ 时，有

$$\| \mathbf{u}^{n+1} - \mathbf{v}^{n+1} \| \to 0, \tag{2.5.11}$$

则称逼近偏微分方程 $Lv=F$ 的差分格式 $L_k^n u_k^n = G_k^n$ 是收敛的。

有些时候，需要讨论差分方程的解收敛到偏微分方程的解的速度。为此，定义收敛的阶 (p,q)。

定义 2.5.3　当 Δx 和 Δt 趋近于 0 时，如果对于任意 t，当 $(n+1)\Delta t$ 收敛到 t 时，有

$$\| \mathbf{u}^{n+1} - \mathbf{v}^{n+1} \| = O(\Delta x^p) + O(\Delta t^q), \tag{2.5.12}$$

则称逼近偏微分方程 $Lv=F$ 的差分格式 $L_k^n u_k^n = G_k^n$ 为阶 (p,q) 收敛。

上述定义中，式(2.5.12)中的高阶无穷小量的严格含义是，存在一个常数 C 满足 $\| \mathbf{u}^{n+1} - \mathbf{v}^{n+1} \| \leqslant C(\Delta x^p + \Delta t^q)$。在这种情况下，常数 C 与时间 t 有关。对于[例 2.5.1]，差分格式(2.5.2)的收敛速度为阶(2,1)。

需要指出，本章将在无限维线性空间讨论并证明初值问题的收敛性(及以后的一致性和稳定性)。我们不要求读者完全掌握泛函分析知识，但只有使用空间 l_∞ 和 $l_{2,\Delta x}$，才能得到初值问题收敛性的准确证明。因此下节将简单介绍无限维线性空间和范的概念。

2.5.2　初边值问题

讨论初边值问题差分格式是否收敛和初值问题格式的差别在于所处的空间。如果使 Δx 或 Δt 较小，问题的空间是序列空间，$-\infty < k < \infty$。如果在区间[0,1]上考虑差分格式，其边界条件为零 Dirichlet 边界条件，那么，成为有限维空间问题。给定 $u_k^0 = f(k\Delta x), k=0,\cdots,M$，$u_0^n = 0, n=1,\cdots$，并且 $u_M^n = 0, n=1,\cdots$，那么，根据差分格式，可有 $u_k^n, k=1,\cdots,M-1; n=1,\cdots$。当 Δx 变小时(或 M 变大)，矢量中的未知量增多。并非不能衡量偏微分方程的解和差分方程解之间的差别，关键在于没有一个"好的空间"，并在此空间内可以收敛。

对于这个问题，有很多解决方法。本文所采用的方法是，让 Δx 趋近于 0，并在一个合适的空间内按某种范数讨论收敛性。

将区间[0,1](当然，可以是其他任意区间)进行分割，使其成为均匀网格，其增量为 Δx。考虑由增量 $\{\Delta x_j\}, j=1,\cdots$ 产生的分割序列，当 $j\to\infty$ 时，$\Delta x_j \to 0$。令 X_j 表示有限维赋范线性空间，它包含了增量 Δx_j 相关的解。如果用 $\| \cdot \|_j$ 表示空间 X_j 上的范数，则可定义初边值问题的收敛性如下。

定义 2.5.4　在时间 t，如果对任意分割序列 $\{\Delta x_j\}$，当 $j\to\infty$，$\Delta t \to 0$ 且 $(n+1)\Delta t \to t$ 时，有

$$\| \mathbf{u}^n - \mathbf{v}^n \|_j \to 0, \tag{2.5.13}$$

则逼近连续初边值问题的差分算法是收敛的。

当然，可以用类似的方式定义收敛阶 (p,q)。

[例 2.5.2]　当 $0 \leqslant r \leqslant 1/2$，证明差分格式

$$u_k^{n+1} = (1-2r)u_k^n + r(u_{k-1}^n + u_{k-1}^n), \quad n \geqslant 0, k = 1,\cdots,M-1; \tag{2.5.14}$$

$$u_0^{n+1} = u_M^{n+1} = 0, \quad n \geqslant 0; \tag{2.5.15}$$

$$u_k^0 = f(k\Delta x), \quad k = 0,\cdots,M, \tag{2.5.16}$$

以范的上界收敛于初始边界值问题

$$\frac{\partial v}{\partial t} = \mu \frac{\partial^2 v}{\partial x^2}, \quad x \in (0,1), t > 0, \tag{2.5.17}$$

$$v(x,0) = f(x), \quad x \in [0,1], \tag{2.5.18}$$

$$v(0,t)=v(1,t)=0,\quad t>0。\tag{2.5.19}$$

[解] 证明差分格式(2.5.14)～(2.5.16)收敛的方法与[例 2.5.1]中初始值问题基本相同，所不同的是本例中矢量是有限长度的，而步长则可以变化。而且，为了证明需添加一些假设。设具有 M_j+1 个点的分割中，Δx_j 是增量，而 X_j 表示具有有限维范的上界

$$\|(u_1,\cdots,u_{M_j-1})^{\mathrm{T}}\|_{M_j-1,\infty}=\sup_{1\leqslant k\leqslant M_j-1}|u_k|\tag{2.5.20}$$

的(M_j-1)维矢量空间。并令

$$z_k^n=u_k^n-v(k\Delta x_j,n\Delta t)。\tag{2.5.21}$$

如[例 2.5.1]，z_k^n 满足

$$z_k^{n+1}=(1-2r)z_k^n+r(z_{k+1}^n+z_{k-1}^n)+O(\Delta t^2)+O(\Delta t\Delta x_j^2),\quad k=1,\cdots,M_j-1。\tag{2.5.22}$$

如果 $0\leqslant r\leqslant 1/2$，方程(2.5.22)右边是非负的，且

$$\begin{aligned}|z_k^{n+1}|&\leqslant(1-2r)|z_k^n|+r|z_{k+1}^n|+r|z_{k-1}^n|+A(\Delta t^2+\Delta t(\Delta x_j)^2)\\&\leqslant\|z^n\|_{M_j-1,\infty}+A(\Delta t^2+\Delta t\Delta x_j^2)。\end{aligned}$$

式中：$z^n=[z_1^n,\cdots,z_{M_j-1}^n]^{\mathrm{T}}$。可得

$$\|z^{n+1}\|_{M_j-1,\infty}\leqslant(n+1)\Delta tA(\Delta t+\Delta x_j^2)。\tag{2.5.23}$$

因此，当 $j\to\infty$，$\Delta t\to 0$ 且$(n+1)\Delta t\to t$ 时，

$$\|u^{n+1}-v^{n+1}\|_{M_j-1,\infty}\to 0,$$

即式(2.5.14)～(2.5.16)收敛。

2.6 线性代数基本概念

有限差分格式的收敛性是在有限维和无限维线性空间上证明的。为了使内容完整，有必要介绍线性代数和泛函分析中的一些空间和范数。若需要了解更详细的内容，请参考文献[6]。

2.6.1 空间和范数

对于有限维空间，如 N 维空间，指的是实欧几里德空间 $\Re^N$ 或复数欧几里德空间 $\mathbf{C}^N$。经常用的有限维范数有：

欧几里德范数：

$$\|u\|_2=\sqrt{\sum_{k=1}^{N}|u_k|^2}。\tag{2.6.1}$$

范数 $l_{2,\Delta x}$：

$$\|u\|_{2,\Delta x}=\sqrt{\sum_{k=1}^{N}|u_k|^2\Delta x}。\tag{2.6.2}$$

范数的上界：

$$\|u\|_\infty=\sup_{1\leqslant k\leqslant N}|u_k|。\tag{2.6.3}$$

需要指出，在空间 $\Re^N$，范数$|\cdot|$表示绝对值，当在空间 $\mathbf{C}^N$ 里面时，它表示复数的幅值。在空间 $\Re^N$ 或 $\mathbf{C}^N$ 中，所有这些范数是等价的。

无限维序列空间包含实和复 l_2 空间，

$$l_2 = \{u = (\cdots, u_{-1}, u_0, u_1, \cdots)^{\mathrm{T}} \mid \sum_{k=-\infty}^{\infty} |u_k|^2 < \infty\}, \tag{2.6.4}$$

其范数为

$$\|u\|_2 = \sqrt{\sum_{k=-\infty}^{\infty} |u_k|^2}。 \tag{2.6.5}$$

实和复 l_2 空间具有能量范数 $l_{2,\Delta x}$，其范数为

$$\|u\|_{2,\Delta x} = \sqrt{\sum_{k=-\infty}^{\infty} |u_k|^2 \Delta x}。 \tag{2.6.6}$$

而所有有界序列空间

$$l_\infty = \{u = (\cdots, u_{-1}, u_0, u_1, \cdots)^{\mathrm{T}} \mid \sup_{-\infty<k<\infty} |u_k| < \infty\}, \tag{2.6.7}$$

其范数为

$$\|u\|_\infty = \sup_{-\infty<k<\infty} |u_k|。 \tag{2.6.8}$$

最后，当进行变换时，将得到复值无限维线性空间，勒贝格平方积分函数 $L_2(\mathbf{R})$，

$$L_2(\mathbf{R}) = \{v: \mathbf{R} \to \mathbf{C} \mid \int_{\mathbf{R}} |v(x)|^2 \mathrm{d}x < \infty\}, \tag{2.6.9}$$

其范数为

$$\|v\|_2 = \sqrt{\int_R |v(x)|^2 \mathrm{d}x}。 \tag{2.6.10}$$

需要指出的是，上面用到的 $\int_{\mathbf{R}}$ 表示在 $\mathbf{R}$ 上进行积分。这种情况下，积分是勒贝格积分。而且，函数 $\mathbf{R}$ 的域一般为实轴 $\Re$，或区间$[-\pi,\pi]$。如果函数为复数时，符号$[v(x)]$指的是复数的幅值。

2.6.2 算子

在上述线性空间中考虑问题时，差分运算及其变换就成了这些空间上的算子。考虑算子 $Q, Q: X \to X$，X 为线性空间，其范数为 $\|\cdot\|$。序列空间上的算子是由第 k 个元素定义的。例如，如果已经定义 Qu^n 的第 k 个元素为$(Qu^n)_k = u_k^n + r\delta^2 u_k^n$，则差分算子 $u_k^{n+1} = u_k^n + r\delta^2 u_k^n$ 可写成 $u^{n+1} = Qu^n$。L_2 空间上的算子一般为函数的乘法，如：$\hat{Q}\hat{u}(\xi) = p(\xi)\hat{u}(\xi)$。定义在有限维空间上的算子(包括差分算子)经常以矩阵的方式描述。

本章会涉及算子的范数，但计算算子的范数几乎不可能。我们将采用的适于计算范数的两个结果都与有限维算子和矩阵有关。因此首先陈述一个结果，它能够计算 Hermitian 矩阵的范数(如果矩阵 $\mathbf{A}^* = \mathbf{A}$，这里 $\mathbf{A}^*$ 是 $\mathbf{A}$ 的共轭转置，则称矩阵 $\mathbf{A}$ 为 Hermitian 矩阵。)

命题 2.6.1 如果矩阵 $\mathbf{A}$ 为 $N \times N$ 的 Hermitian 矩阵，那么

$$\|\mathbf{A}\| = \sigma(\mathbf{A}),$$

式中：$\sigma(\mathbf{A}) = \max\{|\lambda| : \lambda \text{ 为 } A \text{ 的特征值}\}$，$\sigma(\mathbf{A})$称为是矩阵 $\mathbf{A}$ 的谱半径。

同样，对于一般的矩阵，有

命题 2.6.2 如果 $\mathbf{A}$ 为 $N \times N$ 矩阵，那么

$$\|\mathbf{A}\| = \sqrt{\sigma(\mathbf{A}^*\mathbf{A})}。$$

2.6.3 三角矩阵

差分方程所涉及的矩阵大多是三角矩阵。有时需要知道三角矩阵的特征值。对于某些特殊的三角矩阵，其特征值和特征向量是知道的。例如，对于

$$\boldsymbol{T}=\mathrm{Tr}(a,b,c)=\begin{pmatrix} b & c & \cdots & \\ a & b & c\cdots & \\ & \vdots & & \\ \cdots a & b & c & \\ \cdots & a & b & \end{pmatrix}_{N\times N},\tag{2.6.11}$$

其特征值为

$$\lambda_j=b+2c\sqrt{\frac{a}{c}}\cos\frac{j\pi}{N+1}。\tag{2.6.12}$$

对于 $j=1,\cdots,N$，相应的特征向量为

$$\boldsymbol{u}_j=\begin{bmatrix} u_1^{(j)} \\ \vdots \\ u_k^{(j)} \\ \vdots \\ u_N^{(j)} \end{bmatrix},\quad u_k^{(j)}=2\left(\sqrt{\frac{a}{c}}\right)^k\sin\frac{kj\pi}{N+1},\quad k=1,\cdots,N。\tag{2.6.13}$$

其他一些特殊的三角矩阵，其特征值和特征向量为

$$\boldsymbol{T}_{N_1D}=\begin{pmatrix} 1 & -1 & 0 & \cdots & & \\ -1 & 2 & -1 & 0 & & \\ & & \vdots & & & \\ & 0 & -1 & 2 & -1 & \\ & \cdots & 0 & -1 & 2 & \end{pmatrix}_{N\times N},\tag{2.6.14}$$

$$\lambda_j=2-2\cos\frac{(2j-1)\pi}{2N+1},\quad j=1,\cdots,N,\tag{2.6.15}$$

$$u_k^{(j)}=\cos\frac{(2j-1)\pi x_k}{2},\quad k=1,\cdots,N;j=1,\cdots,N。\tag{2.6.16}$$

其中：$x_k=(2k-1)\Delta x/2,k=1,\cdots,N,\Delta x=2/(2N+1)$。

$$\boldsymbol{T}_{N_2D}=\begin{pmatrix} 2 & -2 & 0 & \cdots & & \\ -1 & 2 & -1 & 0 & \cdots & \\ & & \vdots & & & \\ & 0 & -1 & 2 & -1 & \\ & \cdots & 0 & -1 & 2 & \end{pmatrix}_{N\times N},\tag{2.6.17}$$

$$\lambda_j=2-2\cos\frac{(2j-1)\pi}{2N};\quad j=1,\cdots,N,\tag{2.6.18}$$

$$u_k^{(j)}=\cos\frac{(2j-1)\pi x_k}{2},\quad k=1,\cdots,N;j=1,\cdots,N。\tag{2.6.19}$$

其中：$x_k=(k-1)\Delta x,k=1,\cdots,N,\Delta x=1/N$。

$$
\boldsymbol{T}_{N_2N_2}=\begin{bmatrix} 2 & -2 & 0 & \cdots & & \\ -1 & 2 & -1 & 0 & & \\ & & \vdots & & & \\ & & 0 & -1 & 2 & -1 \\ & & \cdots & 0 & -2 & 2 \end{bmatrix}_{N\times N}, \tag{2.6.20}
$$

$$
\lambda_j = 2-2\cos\frac{(2j-1)\pi}{N-1};\quad j=1,\cdots,N, \tag{2.6.21}
$$

$$
u_k^{(j)} = \cos(2j-1)\pi x_k,\quad k=1,\cdots,N;j=1,\cdots,N。 \tag{2.6.22}
$$

其中：$x_k=(k-1)\Delta x, k=1,\cdots,N;\Delta x=1/(N-1)$。

2.7　一致性

2.7.1　初值问题

设考虑的偏微分方程为 $Lv=F$，其相应的有限差逼近为 $L_k^n u_k^n = G_k^n$。其中 G_k^n 表示有限差分逼近的格式。有下述定义。

定义 2.7.1　如果对任意光滑函数 $\phi=\phi(x,t)$，当 $\Delta x,\Delta y\to 0$ 且 $[k\Delta x,(n+1)\Delta t]\to(x,t)$ 时，有

$$
(L\phi-F)\mid_k^n-[L_k^n\phi(k\Delta x,n\Delta t)-G_k^n]\to 0, \tag{2.7.1}
$$

则称有限差分格式 $L_k^n u_k^n=G_k^n$ 与偏微分方程 $Lv=F$ 在点 (x,t) 是点(pointwise)一致的。

如果将差分格式写成

$$
\boldsymbol{u}^{n+1}=Q\boldsymbol{u}^n+\Delta t\boldsymbol{G}^n, \tag{2.7.2}
$$

这里：

$$
\boldsymbol{u}^n=(\cdots,u_{-1}^n,u_0^n,u_1^n,\cdots)^{\mathrm{T}},
$$

$$
\boldsymbol{G}^n=(\cdots,G_{-1}^n,G_0^n,G_1^n,\cdots)^{\mathrm{T}},
$$

而 Q 为相应空间中的算子，那么，关于一致性的更强的定义如下：

定义 2.7.2　当 $\Delta x,\Delta t\to 0$，如果偏微分方程的解 v 满足

$$
\boldsymbol{v}^{n+1}=Q\boldsymbol{v}^n+\Delta t\boldsymbol{G}^n+\Delta t\tau^n, \tag{2.7.3}
$$

这里 $\boldsymbol{v}^n$ 表示向量，其第 k 个分量为 $v(k\Delta x,n\Delta t)$，且

$$
\|\tau^n\|\to 0,
$$

则称有限差分格式(2.7.2)与偏微分方程是按范数 $\|\cdot\|$ 一致的。

下面给出关于一致性的另一个定义，它与上面的定义稍有不同。

定义 2.7.3　如果

$$
\|\tau^n\|=O(\Delta x^p)+O(\Delta t^q), \tag{2.7.4}
$$

则有限差分格式(2.7.2)以精度为阶 (p,q) 逼近相应的偏微分方程。并称 τ^n 或 $\|\tau^n\|$ 为截断误差。

[例 2.7.1]　讨论显式差分格式

$$
\frac{u_k^{n+1}-u_k^n}{\Delta t}=\mu\frac{u_{k+1}^n-2u_k^n+u_{k-1}^n}{\Delta x^2} \tag{2.7.5}
$$

与偏微分方程

$$\frac{\partial v}{\partial t}=\mu\frac{\partial^2 v}{\partial x^2},\quad -\infty<x<\infty;t>0 \tag{2.7.6}$$

的一致性。

[解] 假定偏微分方程(2.7.6)的解为 v,并将 v 代入方程(2.7.5),可见表达式(2.7.5)的左边为零,并有

$$\frac{v_k^{n+1}-v_k^n}{\Delta t}-\frac{\mu}{\Delta x^2}(v_{k+1}^n-2v_k^n+v_{k-1}^n)=O(\Delta t)+O(\Delta x^2)。\tag{2.7.7}$$

根据定义 2.7.1 至定义 2.7.3,必须注意项 $O(\Delta t)+O(\Delta x^2)$。如果 v 对于 t 的二阶导数和对于 x 的四阶导数存在且有界,则表明差分格式(2.7.5)与偏微分方程(2.7.6)是点一致的。

为了证明差分格式(2.7.5)的一致性,或精度为阶(2,1),必须将格式写成式(2.7.2)的形式。为此,将式(2.7.5)的两边乘以 Δt,经过变换得到

$$u_k^{n+1}=u_k^n+\mu\frac{\Delta t}{\Delta x^2}(u_{k+1}^n-2u_k^n+u_{k-1}^n)。\tag{2.7.8}$$

令偏微分方程(2.7.6)的解为 v,记 $r=\mu\Delta t/\Delta x^2$,有

$$\begin{aligned}
\tau_k^n\Delta t &= v_k^{n+1}-[v_k^n+r(v_{k+1}^n-2v_k^n+v_{k-1}^n)]\\
&=v_k^n+\left(\frac{\partial v}{\partial t}\right)_k^n\Delta t+\frac{\partial^2 v}{\partial t^2}(k\Delta x,t_1)\frac{\Delta t^2}{2}\\
&\quad-\left\{v_k^n+r\left[v_k^n+\left(\frac{\partial v}{\partial x}\right)_k^n\Delta x+\left(\frac{\partial^2 v}{\partial x^2}\right)_k^n\frac{\Delta x^2}{2}+\left(\frac{\partial^3 v}{\partial x^3}\right)_k^n\frac{\Delta x^3}{6}\right.\right.\\
&\quad+\left(\frac{\partial^4 v}{\partial x^4}\right)(x_1,n\Delta t)\frac{\Delta x^4}{24}-2v_k^n+v_k^n-\left(\frac{\partial v}{\partial x}\right)_k^n\Delta x\\
&\quad\left.\left.+\left(\frac{\partial^2 v}{\partial x^2}\right)_k^n\frac{\Delta x^2}{2}-\left(\frac{\partial^3 v}{\partial x^3}\right)_k^n\frac{\Delta x^3}{6}+\left(\frac{\partial^4 v}{\partial x^4}\right)(x_2,n\Delta t)\frac{\Delta x^4}{24}\right]\right\}\\
&=\left(\frac{\partial v}{\partial t}\right)_k^n\Delta t-r\Delta x^2\left(\frac{\partial^2 v}{\partial x^2}\right)_k^n+\frac{\partial^2 v}{\partial t^2}(k\Delta x,t_1)\frac{\Delta t^2}{2}\\
&\quad-r\left(\frac{\partial^4 v}{\partial x^4}\right)(x_1,n\Delta t)\frac{\Delta x^4}{24}-r\left(\frac{\partial^4 v}{\partial x^4}\right)(x_2,n\Delta t)\frac{\Delta x^4}{24} \qquad (2.7.9)\\
&=\left(\frac{\partial v}{\partial t}-\mu\frac{\partial^2 v}{\partial x^2}\right)_k^n\Delta t+\frac{\partial^2 v}{\partial t^2}(k\Delta x,t_1)\frac{\Delta t^2}{2}\\
&\quad-\mu\frac{\partial^4 v}{\partial x^4}(x_1,n\Delta t)\frac{\Delta x^4}{24}\Delta t-\mu\frac{\partial^4 v}{\partial x^4}(x_2,n\Delta t)\frac{\Delta x^4}{24}\Delta t, \qquad (2.7.10)
\end{aligned}$$

其中:t_1,x_1 和 x_2 是泰勒级数展开余项中相应的点,而式(2.7.10)是将式(2.7.9)中代入 $r=\mu\Delta t/\Delta x^2$ 后得到的点。因 $\frac{\partial v}{\partial t}-\mu\frac{\partial^2 v}{\partial x^2}=0$,故有

$$\tau_k^n=\frac{\partial^2 v(k\Delta x,t_1)}{\partial t^2}\frac{\Delta t}{2}-\mu\left[\frac{\partial^4 v(x_1,n\Delta t)}{\partial x^4}+\frac{\partial^4 v(x_2,n\Delta t)}{\partial x^4}\right]\frac{\Delta x^4}{24}。\tag{2.7.11}$$

现在决定采用何种范数。如果假设 $\frac{\partial^2 v}{\partial t^2}$ 和 $\frac{\partial^4 v}{\partial x^4}$ 在 $\Re\times[0,t_0]$(对于某个 $t_0>t$)是均匀有界的,称按范数的上界差分格式的精度为阶(2,1)。如果假设 $\frac{\partial^2 v}{\partial t^2}$ 和 $\frac{\partial^4 v}{\partial x^4}$ 对于任意的 Δx 和 Δt 满足

$$\sum_{k=-\infty}^{\infty}\left|\left(\frac{\partial^2 v}{\partial t^2}\right)_k^n\right|^2 \Delta x < A < \infty$$

且

$$\sum_{k=-\infty}^{\infty}\left|\left(\frac{\partial^4 v}{\partial t^4}\right)_k^n\right|^2 \Delta x < B < \infty,$$

那么称按范数 $l_{2,\Delta x}$ 差分格式的精度为阶(2,1)。

2.7.2　初边值问题

如同讨论收敛性时，初边值问题与初值问题不同。在处理一致性时，初边值问题与初值问题也有所不同。虽然点一致性的定义 2.7.1 仍适用于初边值问题，但需要对边界条件进行分析。

如 2.5.2 节，考虑分割区间[0,1]产生的序列，它由空间增量$\{\Delta x_j\}$、相应的空间$\{X_j\}$及其范数$\{\|\cdot\|_j\}$来描述。将方程(2.7.3)和(2.7.4)中的范数替换成范数序列$\|\cdot\|_j$后，定义 2.7.2 和 2.7.3 就适合于初边值问题。初值问题和初边值问题的重要差别在于初边值问题是以可分析的形式给出的，如式(2.7.2)，并且建立在矢量空间的某个逻辑序列上。例如，对于初边值问题的显式差分格式(2.5.2)和 Dirichlet 边界条件，可以容易得到按范数的上界或 $l_{2,\Delta x}$ 范数的点一致性且其精度为阶(2,1)。

由于计算的原因，一致逼近边界条件和一致逼近偏微分方程同样重要。利用一致性得到收敛性时，边界条件的逼近为差分方程的一个组成部分。应该指出，如何处理边界条件是非常重要的。

2.8　稳定性

要证明差分方程的收敛性，需要先证明一致性和稳定性。由于已经讨论了证明一致性和收敛性的方法，本节将讨论稳定性问题。讨论一致性问题所采用的大部分方法可用于讨论稳定性问题。证明收敛性的主要问题是证明稳定性。证明稳定性一般来讲比证明收敛性容易，但对于给定差分格式要证明其稳定性依然比较困难。

2.8.1　初值问题

对于差分格式的稳定性，通常的定义是初始条件微小误差引起差分方程解的误差必须在允许的很小范围内。这个定义允许误差存在，但不得以指数倍增长。可以说，差分算法的稳定性定义类似于偏微分方程的适定性定义。

讨论差分格式

$$\boldsymbol{u}^{n+1} = Q\boldsymbol{u}^n,\quad n \geqslant 0 \tag{2.8.1}$$

的稳定性，它是空间 $\Re$ 上齐次线性偏微分方程初值问题的一般格式。

定义 2.8.1　对于 $0\leqslant t=(n+1)\Delta t$，$0<\Delta x\leqslant\Delta x_0$ 且 $0<\Delta t\leqslant\Delta t_0$，如果存在正常数 Δx_0，Δy_0 和非负常数 K,β，使得

$$\|\boldsymbol{u}^{n+1}\| \leqslant K\mathrm{e}^{\beta t}\|\boldsymbol{u}^0\|, \tag{2.8.2}$$

则差分格式(2.8.1)按范数$\|\cdot\|$是稳定的。

讨论非齐次差分格式的稳定性时，仅考察相应齐次格式的稳定性。非齐次项带来的误差在截断误差项 τ^n 中考虑。文献中有很多类似的稳定性定义，定义 2.8.1 是比较强的定义。有些定义只要求对于任意的 T，有 $(n+1)\Delta t\leqslant T$（其中 K 和 β 与 T 相关），则条件式(2.8.2)成立。更常见的条件是不允许解按指数增长，即将条件式(2.8.2)改为

$$\| u^{n+1} \| \leqslant K \| u^0 \|。\tag{2.8.3}$$

不等式(2.8.2)可以引出关于稳定性的下列特性。

命题 2.8.2　对于 $0\leqslant t=(n+1)\Delta t$，$0<\Delta x\leqslant \Delta x_0$，且 $0<\Delta t\leqslant \Delta t_0$，当且仅当存在正常数 Δx_0，Δt_0 和非负常数 K 和 β，使得

$$\| Q^{n+1} \| \leqslant K e^{\beta t},\tag{2.8.4}$$

则差分格式(2.8.2)是按范数 $\|\cdot\|$ 稳定的。

注意，不等式(2.8.4)中的范数是算子的范数而不是矢量的范数。

[例 2.8.1]　证明差分格式

$$u_k^{n+1} = (1-2r)u_k^n + r(u_{k+1}^n + u_{k-1}^n)\tag{2.8.5}$$

按范数的上界是稳定的。

[解]　本例的解法与[例 2.5.1]相同。注意到如果 $r\leqslant 1/2$，则

$$| u_k^{n+1} | \leqslant (1-2r) | u_k^n | + r(| u_{k+1}^n | + | u_{k-1}^n |) \leqslant \| u^n \|_\infty。$$

将上式的两边对于 k 取上界，得到

$$\| u^{n+1} \|_\infty \leqslant \| u^n \|_\infty,$$

因此满足不等式(2.8.3)，其中 $K=1$ 且 $\beta=0$。

上述证明中要求 $r\leqslant 1/2$，这种情况称差分格式(2.8.5)是条件稳定(conditionally stable)的。其条件是 $r\leqslant 1/2$。如果不需要对 Δt 和 Δx 施加任何约束条件就可以证明给定差分格式的稳定性，则称该差分格式是稳定(stable)的或无条件稳定(unconditionally stable)的。

2.8.2　初边值问题

如同在收敛性和稳定性问题中所考虑的，假定由区间划分所产生的序列，此序列由增量 $\{\Delta x_j\}$，序列的空间 $\{X_j\}$ 和其范数 $\|\cdot\|_j$ 来描述。当它满足不等式(2.8.2)或(2.8.3)时，此时范数为 $\|\cdot\|_j$，那么，称初边值问题的差分格式是稳定的。

[例 2.8.2]　证明初边值问题

$$\frac{\partial v}{\partial t} = \mu \frac{\partial^2 v}{\partial x^2},\quad x\in(0,1), t>0,\tag{2.8.6}$$

$$v(x,0) = f(x),\quad x\in[0,1],\tag{2.8.7}$$

$$v(0,t) = v(1,t) = 0,\quad t>0\tag{2.8.8}$$

的差分格式

$$u_k^{n+1} = u_k^n + r\delta^2 u_k^n,\quad k=1,\cdots,M-1,\tag{2.8.9}$$

$$u_0^{n+1} = u_M^{n+1} = 0,\tag{2.8.10}$$

$$u_k^0 = f(k\Delta x),\quad k=0,\cdots,M。\tag{2.8.11}$$

如果 $r\leqslant 1/2$，差分格式(2.8.9)～(2.8.11)是稳定的。

[解]　设空间 X_j 是 (M_j-1) 个矢量张成的空间，其中 $M_j\Delta x_j=1$，并且范数 $\|\cdot\|_j$ 表示在空间 X_j 上范数的上界。

注意到如果 $r\leqslant 1/2$，有

$$\begin{aligned}|u_k^{n+1}| &= |(1-2r)u_k^n + r(u_{k+1}^n + u_{k-1}^n)| \\ &\leqslant |(1-2r)||u_k^n| + r(|u_{k+1}^n| + |u_{k-1}^n|) \\ &\leqslant (1-2r)\|u^n\|_j + r\|u^n\|_j + r\|u^n\|_j \\ &= \|u^n\|_j。\end{aligned}$$

两边对 k 取上界，得

$$\|u^{n+1}\|_j \leqslant \|u^n\|_j。$$

重复使用上式，得到

$$\|u^{n+1}\|_j \leqslant \|u^0\|_j。$$

因此当 $K=1$ 且 $\beta=0$，差分格式是稳定的。

2.9 Lax 定理

我们已经讨论了收敛性、一致性和稳定性，Lax 定理则给出它们之间的联系。

定理 2.9.1 Lax 等价定理　对于适定的线性初值问题，如果差分格式

$$u^{n+1} = Qu^n + \Delta t G^n \tag{2.9.1}$$

与原方程具有一致性，那么当且仅当该格式是稳定的时候，这个差分格式是收敛的。

就是说，只要差分格式与原方程具有一致性，那么这个差分格式的收敛性和稳定性是等价的。我们不对此定理进行证明。

定理 2.9.2 Lax 定理　对于适定的线性初值问题，如果差分格式

$$u^{n+1} = Qu^n + \Delta t G^n$$

在范数 $\|\cdot\|$ 意义下，其精度的阶为 (p,q)，而且按范数 $\|\cdot\|$ 来说，它是稳定的，那么称按范数 $\|\cdot\|$ 这个差分格式以阶 (p,q) 收敛。

[证明]　令 $v=v(x,t)$ 表示初值问题的精确解。那么，由于差分格式的精度为阶 (p,q)，有

$$v^{n+1} = Qv^n + \Delta t G^n + \Delta t \tau^n。 \tag{2.9.2}$$

这里 $\|\tau^n\| = O(\Delta x^p) + O(\Delta t^q)$。定义 w^j 为 $v^j - u^j$，那么 w 满足

$$w^{n+1} = Qw^n + \Delta t \tau^n。 \tag{2.9.3}$$

反复应用方程(2.9.3)，得到

$$\begin{aligned}w^{n+1} &= Qw^n + \Delta t\tau^n \\ &= Q(Qw^{n-1} + \Delta t\tau^{n-1}) + \Delta t\tau^n \\ &= Q^2 w^{n-1} + \Delta t Q\tau^{n-1} + \Delta t\tau^n \qquad (2.9.4)\\ &\;\;\vdots \\ &= Q^{n+1}w^0 + \Delta t\sum_{j=0}^{n} Q^j\tau^{n-j}, \qquad (2.9.5)\end{aligned}$$

$$w^{n+1} = \Delta t\sum_{j=0}^{n} Q^j\tau^{n-j}。 \tag{2.9.6}$$

由于差分格式是稳定的，因此，对于任意 j，有

$$\|Q^j\| \leqslant Ke^{\beta t}。 \tag{2.9.7}$$

对方程(2.9.6)两边取范数并应用式(2.9.7)，得到

$$\begin{aligned}\| w^{n+1} \| &\leqslant \Delta t \sum_{j=0}^{n} \| Q^j \| \| \tau^{n-j} \| \\ &\leqslant \Delta t K \sum_{j=0}^{n} e^{\beta j \Delta t} \| \tau^{n-j} \| \\ &\leqslant \Delta t K e^{\beta(n+1)\Delta t} \sum_{j=0}^{n} \| \tau^{n-j} \| \\ &\leqslant (n+1)\Delta t K e^{\beta(n+1)\Delta t} C^*(t)(\Delta x^p + \Delta t^q)。\end{aligned} \tag{2.9.8}$$

式中：$C^*(t)=\sup_{0\leqslant s\leqslant t}C(s)$，这里 $C(s)$，$s=(n-j)\Delta t$。正如在收敛性定义中那样，t 的选择为：当 $\Delta t\to 0$ 时，$(n+1)\Delta t\to t$。因此，当 $\Delta x,\Delta t\to 0$ 时，得到式(2.9.8)。

$$(n+1)\Delta t K e^{\beta(n+1)\Delta t} C^*(t)(\Delta x^p + \Delta t^q) \to t K e^{\beta t} C^*(t) 0 = 0 \tag{2.9.9}$$

因此，它等价于 $\| u^{n+1} - v^{n+1} \| \to 0$。

为了得到阶(p,q)收敛，注意到式(2.9.8)可重新写为

$$\begin{aligned}\| w^{n+1} \| &\leqslant K(t)(\Delta x^p + \Delta t^q) \\ &= O(\Delta x^p) + O(\Delta t^q)。\end{aligned}$$

这里，有几点是必须提醒的：

(1) 式(2.9.1)和(2.9.2)中项 G^n 是对原偏微分方程的逼近，其逼近误差由截断误差项 τ 度量。

(2) 证明一致性和稳定性必须使用同一范数，那么按这个范数，差分格式是收敛的。

(3) 仅考虑点一致是不充分的。如果 $r\leqslant 1/2$，显式差分格式(2.8.5)按范数的上界以阶(2,1)收敛。

下面讨论初边值问题。

假定初边值问题具有与在收敛性、一致性和稳定性中处理初边值问题时同样的设定，那么，很容易看到，定理 2.9.2 的证明在这里同样适用，只是对每个范数加一个下标 j。因此，有下面的定理。

定理 2.9.3 Lax 定理 对于适定线性初边值问题，如果差分格式(2.9.1)在范数序列 $\|\cdot\|_j$ 意义下，其精度的阶为(p,q)，而且对于范数序列 $\|\cdot\|_j$，差分格式是稳定的，那么称按范数序列 $\|\cdot\|_j$，这个差分格式以阶(p,q)收敛。

实际上对于初边值问题，偏微分方程和边界条件都要离散化成

$$u^{n+1} = Qu^n + \Delta t G^n$$

的形式是非常困难的。

参 考 文 献

[1] Thomas J W. Numerical Partial Differential Equations-Finite Difference Methods. New York: Springer-Verlag, 1995

[2] Isaacson E, Keller H B. Analysis of Numerical Methods. New York: Dover Publication, Inc., 1994

[3] Richtmyer R D, Morton K W. Difference Methods for Initial-Value Problems. New York: Interscience Publishers, 1967

[4] Sod G A. Numerical Methods in Fluid Dynamics. Cambridge: Cambridge University Press, 1985

[5] Strikwerda J C. Finite Difference Schemes and Partial Differential Equations. California: Pacific Grove, 1989

[6] Taylor A E. Introduction Functional Analysis. New York: John Wiley & Sons Inc., 1957

[7] Quarteroni A, Valli A. Numerical Approximation of partial differential equations. New York: Springer-Verlag, 1997

[8] Isaacson E, keller H B. Analysis of Numerical Methods. New York: John Wiley & Sons, 1966

[9] 吕涛,石济民,林振宝. 区域分解算法——偏微分方程数值解新技术. 北京:科学出版社,1999

[10] 张宝林,袁国兴,刘兴平等. 偏微分方程并行有限差分方法. 北京:科学出版社,1994

[11] 李德元,陈光南. 抛物型方程差分方法引论. 北京:科学出版社,1995

[12] 李伟. 粘性流体的混合有限分析解法. 北京:科学出版社,2000

第3章　线性代数方程的数值解法

求线性代数方程的数值解是科学计算的核心，也是有限差分法解偏微分方程的一个重要步骤。本章介绍线性代数方程（组）的常用解法，并给出线性代数方程（组）几种迭代方法。这些迭代方法的加速参数不需依据特征值分析来确定，而是可以直接根据递推关系导出，避免了计算工作量较大的估算特征值问题。

3.1　直接解法

考虑实系数代数方程组

$$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b},\tag{3.1.1}$$

式中：系数矩阵 $\boldsymbol{A}$ 是非奇异的 n 维矩阵。直接法能得到式(3.1.1)的精确解。

3.1.1　高斯消元法

最经典的直接法就是高斯消元法（Gaussian Elimination method，简称 GEM），高斯消元法将系数矩阵 $\boldsymbol{A}$ 分解成 $\boldsymbol{LU}$ 的积，其中，$\boldsymbol{L}$ 和 $\boldsymbol{U}$ 分别是上三角和下三角矩阵。分解过程可以有 n 步：

$$\boldsymbol{A}=\boldsymbol{A}^{(1)}\rightarrow\boldsymbol{A}^{(2)}\rightarrow\cdots\rightarrow\boldsymbol{A}^{(n)}=\boldsymbol{U}=L^{-1}\boldsymbol{A}。\tag{3.1.2}$$

高斯消元法的主要思路是，矩阵 $\boldsymbol{A}^{k}$ 变换到 $\boldsymbol{A}^{(k+1)}$ 的过程中使新矩阵 $\boldsymbol{A}^{(k+1)}$ 的对角线下第 k 列元素为零。具体算法的过程为：

$$\begin{aligned}&\text{For}\quad k=1,\cdots,n-1\\&\quad\text{For}\quad i=k+1,\cdots,n\\&\qquad m_{ik}=a_{ik}^{(k)}/a_{kk}^{(k)}\\&\qquad\text{For}\quad j=k+1,\cdots,n\\&\qquad\quad a_{ij}^{(k+1)}=a_{ij}^{(k)}-m_{ik}a_{kj}^{(k)}\\&\qquad\text{end}\quad j\\&\quad\text{end}\quad i\\&\text{end}\quad k\end{aligned}\tag{3.1.3}$$

式中：$a_{ij}^{(1)}$ 是经过一次变换后的矩阵 $\boldsymbol{A}=\boldsymbol{A}^{(1)}$ 的元素，m_{ik} 是矩阵 $\boldsymbol{L}$ 非对角线下的元素。而且，当 $i=1,\cdots,n$ 时，矩阵 $\boldsymbol{L}$ 的对角线上元素 $l_{ii}=1$。这个算法需要 $O(n^3)$ 次运算。高斯消元法的问题是，只要矩阵 $\boldsymbol{A}$ 主对角线上元素 $a_{kk}^{(k)}$ 接近于零，这个算法可能不稳定。

3.1.2　*LU* 分解

高斯消元法不稳定的问题可以用对角线元素重组方法解决。其思路是，在每次消元的步骤 k，将后 $n-k$ 列或后 $n-k$ 行，与后 $n-k$ 行或 $n-k$ 列进行交换，以使对角线元素 $\tilde{a}_{kk}^{(k)}$ 的绝对值最大。就是说在每次消元的步骤 k 中，将矩阵 $\boldsymbol{A}^{(k)}$ 依次乘以两个置换矩阵，得到

$$\tilde{A}^{(k)} = P_k A^{(k)} \Pi_k,$$

然后再进行消元处理。这是对于置换矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{\Pi}$ 的完全对角线元素占优分解处理 $\boldsymbol{PA\Pi}=\boldsymbol{LU}$。实际计算中常采用部分对角线占优处理，就是说将矩阵 $\boldsymbol{A}$ 分解成 $\boldsymbol{QA}=\boldsymbol{LU}$，其中，$\boldsymbol{Q}$ 是另一个置换矩阵。

部分对角线占优处理需要在矩阵的元素间进行 n^2 次比较，而完全对角线占优处理需要进行 $2n^3/3$。就是说当 n 比较大时，部分对角线占优处理需要的计算时间相对比较短。

从式(3.1.2)可知，高斯消元法将原始矩阵分解成 $\boldsymbol{A}=\boldsymbol{LU}$。这样，解式(3.1.1)成为解方程组 $\boldsymbol{Ly}=\boldsymbol{b}$ 和 $\boldsymbol{Ux}=\boldsymbol{y}$。由于 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 是三角阵，求 $\boldsymbol{Ly}=\boldsymbol{b}$ 称为向前代换，而得到最终结果 $\boldsymbol{Ux}=\boldsymbol{y}$，则称回代求解。这些运算的计算复杂度为 $O(n^2)$。

注意到，对于 $k=1,\cdots,n$，$l_{kk}=1$。因子 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 可直接由下列算法得到：

$$
\begin{aligned}
&\text{For} \quad k = 1,\cdots,n \\
&\quad \text{For} \quad j = k,\cdots,n \\
&\qquad u_{kj} = a_{kj} - \sum_{p=1}^{k-1} l_{kp} u_{pj} \\
&\quad \text{end} \quad j \\
&\quad \text{For} \quad i = k+1,\cdots,n \quad \text{and} \quad k \neq n \\
&\qquad l_{ik} = \frac{1}{u_{kk}}\left\{a_{ik} - \sum_{p=1}^{k-1} l_{ip} u_{pk}\right\} \\
&\quad \text{end} \quad i \\
&\text{end} \quad k
\end{aligned}
\tag{3.1.4}
$$

仿此，可将 $\boldsymbol{A}=\boldsymbol{LU}$ 分解，其中，矩阵 $\boldsymbol{U}$ 的对角元素 u_{kk} 为 1。如果，矩阵 $\boldsymbol{A}$ 的所有主导子模块是非奇异的，则存在两个下三角矩阵 $\boldsymbol{L}$ 和 $\boldsymbol{M}$，其对角元素都等于 1，且有对角矩阵 $\boldsymbol{D}$，使

$$\boldsymbol{A} = \boldsymbol{LDM}^{\mathrm{T}}, \tag{3.1.5}$$

式中：$\boldsymbol{M}^{\mathrm{T}}$ 是矩阵 $\boldsymbol{M}$ 的转置矩阵。完成这个分解后，只要解三个特殊方程组

$$\boldsymbol{Ly} = \boldsymbol{b}, \quad \boldsymbol{Dz} = \boldsymbol{y}, \quad \boldsymbol{M}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{z}。\tag{3.1.6}$$

如果 $\boldsymbol{A}$ 是对称的，即 $\boldsymbol{M}=\boldsymbol{L}$，那么，式(3.1.5)成为

$$\boldsymbol{A} = \boldsymbol{LDL}^{\mathrm{T}}。\tag{3.1.7}$$

如果对于所有 $\boldsymbol{W}\in\mathfrak{R}^n$，$\boldsymbol{W}\neq 0$，方阵满足

$$(\boldsymbol{Aw},\boldsymbol{w}) > 0, \tag{3.1.8}$$

则称为是正定的(简记，$A>0$)。式中，$(\cdot,\cdot)$是 $\mathfrak{R}^n$ 空间中欧氏标量积。这里要提请读者注意，$\boldsymbol{A}>0$ 有时也指矩阵是正实，这时矩阵 $\boldsymbol{A}$ 是对称的。

如果矩阵 $\boldsymbol{A}$ 是正定对称的，Sylvester 定理证明，其所有主导矩阵是非奇异的，而且矩阵 $\boldsymbol{A}$ 满足下列 Cholesky 分解：

$$\boldsymbol{A} = \boldsymbol{LL}^{\mathrm{T}}, \tag{3.1.9}$$

式中：矩阵 $\boldsymbol{L}$ 是下三角矩阵且其所有对角线上元素为正，这个分解是唯一的。矩阵 $\boldsymbol{L}$ 的元素可以用下列算法得到：

$$
\begin{aligned}
&\text{For} \quad k = 1,\cdots,n \\
&\qquad l_{kk} = \left\{a_{kk} - \sum_{p=1}^{k-1} l_{kp}^2\right\}^{1/2}
\end{aligned}
$$

$$\text{For} \quad i = k+1, \cdots, n \text{ and } k \neq n \tag{3.1.10}$$

$$l_{ik} = \frac{1}{l_{kk}}\left[a_{ik} - \sum_{p=1}^{k-1} l_{ip} l_{kp}\right]$$

$$\text{end} \quad i$$

$$\text{end} \quad k$$

这个算法需要 $O(n^3)$操作(只有高斯消元法的一半),加上 n 次平方根运算。

3.1.3 *QR* 分解

对于通常矩阵 $\boldsymbol{A}$,另一种分解称为 $\boldsymbol{QR}$ 分解。它的算法比 $\boldsymbol{LU}$ 分解复杂些,但更稳定。对于任意实矩阵 $A \in \Re^{m\times n}$(即,m 行 n 列),且 $m \geqslant n$,它的秩为 n,则存在唯一的矩阵 $Q \in \Re^{m\times n}$满足

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = \boldsymbol{D}, \boldsymbol{D} = \mathrm{diag}(d_1, \cdots, d_n), d_k > 0, \text{对于 } k = 1, \cdots, n, \tag{3.1.11}$$

和上三角矩阵 $\boldsymbol{R} \in \Re^{m\times n}$,其 $r_{kk} = 1, k = 1, \cdots, n$,使

$$\mathbf{A} = \boldsymbol{QR}。 \tag{3.1.12}$$

这个分解对于最小二乘方问题特别有效。注意到式(3.1.12)的赋范方程

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}\boldsymbol{x} = \mathbf{A}^{\mathrm{T}}\boldsymbol{b}, \tag{3.1.13}$$

可以导出

$$0 = \mathbf{A}^{\mathrm{T}}(\boldsymbol{b} - \mathbf{A}\boldsymbol{x}) = \boldsymbol{R}^{\mathrm{T}}\boldsymbol{Q}^{\mathrm{T}}(\boldsymbol{b} - \mathbf{A}\boldsymbol{x}) = \boldsymbol{R}^{\mathrm{T}}(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b} - \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{QR}\boldsymbol{x}),$$

因此,有

$$\boldsymbol{D}\boldsymbol{y} = \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{b}, \quad \boldsymbol{R}\boldsymbol{x} = \boldsymbol{y}。 \tag{3.1.14}$$

这样,计算式(3.1.13)仅需要解两个方程组,一个方程的系数矩阵是对角线阵,另一个是上三角阵。

而且,计算因子 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 有很多算法,例如:改进的 Gram-Schmidt 正交化算法和 Household 法。

3.1.4 带状方程组

在 2.6.3 节中给出一些特殊的三角矩阵的特征值和特征向量的计算公式。本章中方程式(3.1.1)是偏微分方程经过离散化得到的,矩阵 $\boldsymbol{A}$ 常是带状的。就是说,存在一个整数 q(带宽),$1 \leqslant q < n$,满足

$$a_{ij} = 0, \quad \text{当 } i,j \text{ 为 } |i-j| > q。 \tag{3.1.15}$$

带状矩阵 $\boldsymbol{A}$ 每行最多有 $2q+1$ 个非零元素。

解带状方程组时,矩阵 $\boldsymbol{A}$ 分解后的三角形矩阵也是带状的,因此可以节省很多计算量和存储空间。自然,如果矩阵 $\boldsymbol{A}$ 是带宽为 q 的带状阵,且能分解为 $\boldsymbol{LU}$,那么,矩阵 $\boldsymbol{L}$ 和 $\boldsymbol{U}$ 也是带状的,且其带宽都为 q。对于有带状矩阵 $\boldsymbol{A}$ 组成的方程组,直接用改进的高斯消元法(3.1.3)计算,需要 $O(nq^2)$次运算。而用向前代换和回代求解,则需要 $O(nq)$次运算。

通常,高斯消元法需要交换矩阵 $\boldsymbol{A}$ 的行和列。这样,带状矩阵 $\boldsymbol{A}$ 的结构会发生变化,矩阵的带宽 q 也会变化。

如果带状矩阵 $\boldsymbol{A}$ 是对称正定的,且满足(3.1.15),它的 Cholesky 分解式(3.1.9)为

$$\text{For} \quad k = 1, \cdots, n$$

$$s_k = \max(1, k-q), r_k = \min(n, k+q)$$

$$l_{kk} = \left\{a_{kk} - \sum_{p=s_k}^{k-1} l_{kp}^2\right\}^{1/2}$$

For $i = k+1,\cdots,r_k$ and $k \neq n$

$$l_{ik} = \frac{1}{l_{kk}}\left\{a_{ik} - \sum_{p=s_i}^{k-1} l_{ip} l_{kp}\right\}$$

end i

end k

需要 $O(nq^2)$次运算,加上 n 次平方根运算。

3.2 迭代方法概述

一般来说,迭代方法都是将矩阵 $\boldsymbol{A}$ 适当地分成两部分:

$$\boldsymbol{A} = \boldsymbol{P} - \boldsymbol{N}, \tag{3.2.1}$$

其中:P 是非奇异的。那么序列$\{\boldsymbol{x}_k\}$的迭代公式为

$$\boldsymbol{P}\boldsymbol{x}^{k+1} = \boldsymbol{N}\boldsymbol{x}^k + \boldsymbol{b}, \quad k \geqslant 0, \tag{3.2.2}$$

其中:$\boldsymbol{x}^0$ 是给定的迭代初始值。应该指出的是,以后讨论的迭代方法之间的差别只是选取不同的 $\boldsymbol{P}$ 和 $\boldsymbol{N}$。

评价一个迭代公式的性能,须考察两个方面:

(1) 迭代公式的收敛速度。

(2) 每步的计算量。

对于收敛速度,设

$$\boldsymbol{e}^k = \boldsymbol{x}^k - \boldsymbol{x}, \quad \boldsymbol{B} = \boldsymbol{P}^{-1}\boldsymbol{N}, \tag{3.2.3}$$

从式(3.2.2)可导出误差方程

$$\boldsymbol{e}^k = \boldsymbol{B}\boldsymbol{e}^{k-1} = \boldsymbol{B}^k\boldsymbol{e}^0, \tag{3.2.4}$$

其中:$\boldsymbol{e}^0$ 是初始误差。

对于任意初值 $\boldsymbol{x}^0$,迭代序列 $\boldsymbol{x}^k$ 收敛于式(3.1.1)的解 $\boldsymbol{x}$ 的充要条件是迭代矩阵 $\boldsymbol{B}$ 的谱半径满足

$$\rho(\boldsymbol{B}) < 1。 \tag{3.2.5}$$

实际上,对于每个 $\varepsilon>0$,可以找到矩阵的范数 $\|\cdot\|$,使 $\|\boldsymbol{B}\| \leqslant [\rho(\boldsymbol{B})+\varepsilon]^k |\boldsymbol{e}^0|$,如果 ε 足够小,那么是收敛的。相反,如果 $\rho(\boldsymbol{B})\geqslant 1$,只有选择 $\boldsymbol{x}^0$ 为 $e^0 = \omega^*$,即特征矢量对应于绝对值最大的特征值 λ。这样,式(3.2.4)成为

$$|\boldsymbol{e}^k| = |\boldsymbol{B}^k\boldsymbol{e}^0| = |\lambda|^k |\boldsymbol{e}^0| = [\rho(\boldsymbol{B})]^k |\boldsymbol{e}^0|,$$

由于误差不会趋于零,因此不收敛。很明显,$\rho(\boldsymbol{B})$越小,收敛速度就越快。

参数 $\sigma = -\log\rho(\boldsymbol{B})$ 称为迭代公式(3.2.2)的渐进收敛率(asymptotic rate of convergence)。

定理 3.2.1 (Householder-John)设矩阵 $\boldsymbol{A}$ 是对称、正定的,并能分解成式(3.2.1)。假设矩阵 $\boldsymbol{P}+\boldsymbol{P}^{\mathrm{T}}-\boldsymbol{A}$ 是正定的,那么

$$\rho(\boldsymbol{P}^{-1}\boldsymbol{N}) < 1。$$

这是迭代方法收敛的一个充分条件。

定理 3.2.2 收敛条件(3.2.5)成立的充分必要条件是，$I-B$ 是非奇异的，并且

$$H=(I-B)^{-1}(I+B) \tag{3.2.6}$$

的所有特征值都有正实部[1]。

3.3 Jacobi 算法

Jacobi 算法是最简单的迭代方法，可用于对角元素非零的矩阵。如果矩阵 A 能表示成

$$A=\begin{bmatrix} & & U_A \\ & D_A & \\ L_A & & \end{bmatrix}, \tag{3.3.1}$$

式中：D_A 是矩阵 A 的对角阵，而 L_A 和 U_A 分别是矩阵 A 的上三角和下三角部分。

设

$$D=\begin{bmatrix} & & 0 \\ & D_A & \\ 0 & & \end{bmatrix},\quad E=\begin{bmatrix} & & 0 \\ & 0 & \\ L_A & & \end{bmatrix},\quad F=\begin{bmatrix} & & U_A \\ & 0 & \\ 0 & & \end{bmatrix},$$

Jacobi 法基于下列分解：

$$P=D,\quad N=-(E+F), \tag{3.3.2}$$

因此，Jacobi 迭代矩阵为

$$B_J=-D^{-1}(E+F)。$$

定义残差矢量为

$$r^k=b-Ax^k, \tag{3.3.3}$$

Jacobi 迭代方程可以写成

$$x^{k+1}-x^k=D^{-1}r^k。$$

Jacobi 迭代算法用矩阵元素形式表示为

$$x_i^{k+1}=\frac{1}{a_{ii}}\left(b_i-\sum_{\substack{j=1\\j\neq i}}^{n}a_{ij}x_j^k\right),\quad i=1,\cdots,n。 \tag{3.3.4}$$

如果矩阵 A 是严格对角占优，那么满足条件 $\rho(B_J)<1$。实际上，

$$\rho(B_J)\leqslant\|D^{-1}(E+F)\|_\infty=\max_{1\leqslant i\leqslant n}\sum_{\substack{j=1\\j\neq i}}\left|\frac{a_{ij}}{a_{ii}}\right|<1,$$

而且，矩阵 A 的对角占优越明显，收敛的速度就越快。

3.4 Gauss-Seidel 法

Gauss-Seidel 法基于下列分解：

$$P=D+E,\quad N=-F, \tag{3.4.1}$$

同 Jacobi 法一样，也是对角线上元素非零时有效。对应的迭代矩阵是

$$B_{GS}=-(D+E)^{-1}F。$$

迭代式(3.2.2)可以写成

$$x_i^{k+1}=\frac{1}{a_{ii}}\left[b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{k+1}-\sum_{j=i+1}^{n}a_{ij}x_j^{k}\right],\quad i=1,\cdots,n。\tag{3.4.2}$$

如果矩阵 $\boldsymbol{A}$ 是对称正定的，那么 Gauss-Seidel 法是收敛的。这可以根据定理 3.2.1 得到。实际上，$\boldsymbol{E}^{\mathrm{T}}=\boldsymbol{F}$ 且

$$\boldsymbol{P}+\boldsymbol{P}^{\mathrm{T}}-\boldsymbol{A}=(\boldsymbol{D}+\boldsymbol{E})+(\boldsymbol{D}+\boldsymbol{E})^{\mathrm{T}}-(\boldsymbol{D}+\boldsymbol{E}+\boldsymbol{F})=\boldsymbol{D},\tag{3.4.3}$$

而 $\boldsymbol{D}$ 是正定的。

Gauss-Seidel 法收敛的另一种情况是矩阵 $\boldsymbol{A}$ 是严格对角占优的。

3.5 松弛算法(S.O.R.和S.S.O.R.)

超松弛算法(Succcessive Over-Relaxation (S.O.R.))是用一个正参数 ω 对 Gauss-Seidel 法迭代加速。假设对于 $i=1,\cdots,n,a_{ii}\neq 0$，为了从 $\boldsymbol{x}^k$ 得到新的迭代式 $\boldsymbol{x}^{k+1}$，有

$$x_i^{k+1}=(1-\omega)x_i^k+\frac{\omega}{a_{ii}}\left[b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{k+1}-\sum_{j=i+1}^{n}a_{ij}x_j^{k}\right]。\tag{3.5.1}$$

其中 $k\geqslant 0,i=1,\cdots,n$，就是说用下列分解：

$$\boldsymbol{P}=\frac{\boldsymbol{D}}{\omega}+\boldsymbol{E},\quad \boldsymbol{N}=\left(\frac{1}{\omega}-1\right)\boldsymbol{D}-\boldsymbol{F}。\tag{3.5.2}$$

对应的迭代矩阵为

$$\boldsymbol{B}_\omega=(\boldsymbol{D}+\omega\boldsymbol{E})^{-1}[(1-\omega)\boldsymbol{D}-\omega\boldsymbol{F}]。\tag{3.5.3}$$

如果 $\omega=1$，就是 Gauss-Seidel 法的 $\boldsymbol{B}_{\mathrm{GS}}$。注意，即使矩阵 $\boldsymbol{A}$ 是对称的，矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{B}_\omega$ 也不一定对称。

关于收敛问题，对于 ω 有下列必要条件：

定理 3.5.1 (Kahan) S.O.R. 迭代矩阵需要满足

$$\rho(\boldsymbol{B}_\omega)\geqslant|\omega-1|,\tag{3.5.4}$$

因此，收敛的必要条件是

$$0<\omega<2。\tag{3.5.5}$$

一般地，当 $0<\omega\leqslant 1$ 时，S.O.R. 法收敛。然而，对于一类特殊矩阵，条件(3.5.5)是收敛的充分条件。

定理 3.5.2 (Ostrowski-Reich)如果矩阵 $\boldsymbol{A}$ 是对称正定的，S.O.R. 法收敛的充分必要条件是满足(3.5.5)。

S.O.R. 法的一个对称改进算法为，在计算 x^k 到 x^{k+1} 的过程中，先计算 S.O.R. 迭代式(3.5.1)，其结果用 y^{k+1} 表示。然后用逆向 S.O.R. 迭代，得到

$$x_i^{k+1}=(1-\omega)y_i^{k+1}+\frac{\omega}{a_{ii}}\left[b_i-\sum_{j=1}^{i-1}a_{ij}y_i^{k+1}-\sum_{j=i+1}^{n}a_{ij}x_j^{k+1}\right]。\tag{3.5.6}$$

其中：对于 $i=n,\cdots,1$，这个算法称为对称 S.O.R. 法(S.S.O.R.)，它是基于下列分解：

$$\boldsymbol{P}=\frac{1}{\omega(2-\omega)}(\boldsymbol{D}+\omega\boldsymbol{E})\boldsymbol{D}^{-1}(\boldsymbol{D}+\omega\boldsymbol{F}),\tag{3.5.7}$$

$$\boldsymbol{N}=\frac{1}{\omega(2-\omega)}[(1-\omega)\boldsymbol{D}-\omega\boldsymbol{E}]\boldsymbol{D}^{-1}[(1-\omega)\boldsymbol{D}-\omega\boldsymbol{F}]。$$

其迭代矩阵为

$$B_\omega^S = (D+\omega F)^{-1}D(D+\omega E)^{-1}\times[(1-\omega)D-\omega E]D^{-1}[(1-\omega)D-\omega F]。\quad (3.5.8)$$

3.6 切比雪夫(Chebyshev)加速法

用切比雪夫多项式也可以加速迭代式(3.2.2)的收敛。其基本思路是，将序列$\{x^k\}$改为

$$y^m = \sum_{k=0}^{m}\gamma_k^m x^k, \quad (3.6.1)$$

其中：系数 γ_k^m 需要根据一个最优指标决定。当 $x^k=x$ 时，对于所有的 $k=0,\cdots,m$，要满足 $y^m=x$，首先要求 $\sum_{k=0}^{m}\gamma_k^m=1$，特别是，$y^0=x^0$。

如果设

$$\varepsilon^k = y^k - x,\ P_m(t)=\sum_{k=0}^{m}\gamma_k^m t^k,$$

得到递推公式

$$\varepsilon^m = P_m(B)\varepsilon^0, \quad (3.6.2)$$

式中：B 是式(3.2.3)中的迭代矩阵。现在的问题是，寻找满足 $P_m(1)=1$，且多项式的次数小于或等于 m 的切比雪夫多项式 P_m，使它的谱半径 $P_m(B)$最小。

如果 B 是对称的并且它的实特征值λ_i 满足

$$-1<\alpha\leqslant\lambda_1\leqslant\cdots\leqslant\lambda_n\leqslant\beta<1,\quad \alpha<\beta,$$

就是说，$P_m(B)$也是对称的并且它的特征值为 $P_m(\lambda_i)$，因此

$$\rho(P_m(B)) = \max_{1\leqslant i\leqslant n}|P_m(\lambda_i)|\leqslant\max_{\alpha\leqslant\lambda\leqslant\beta}|P_m(\lambda)|。\quad (3.6.3)$$

在所有满足 $P_m(1)=1$，且项数小于或等于 m 的切比雪夫多项式 P_m，经过优化使不等式(3.6.3)成立，有

$$P_m(t)=\frac{T_m\left(\frac{\beta+\alpha-2t}{\beta-\alpha}\right)}{T_m\left(\frac{\beta+\alpha-2}{\beta-\alpha}\right)},$$

式中：$T_m(x)$是切比雪夫多项式。

根据式(3.6.2)和(3.6.3)有

$$|\varepsilon^m|_2\leqslant\|P_m(B)\|_2|\varepsilon^0|_2\leqslant\left|T_m\left(\frac{\beta+\alpha-2}{\beta-\alpha}\right)\right|^{-1}|\varepsilon^0|_2, \quad (3.6.4)$$

其中：

$$|v|=|v|_2=(v,v)^{1/2}=\left(\sum_{i=1}^{n}v_i^2\right)^{1/2},\quad v\in\Re^n,$$

表示矢量 v 的长度。

切比雪夫加速法可以按下列步骤计算：

$$y^{m+1}=\omega_{m+1}(y^m-y^{m-1}+\gamma z^m)+y^{m-1},\quad m\geqslant1,$$

其中：

$$\boldsymbol{P}\boldsymbol{z}^m = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{y}^m, \quad \omega_{m+1} = 2\,\frac{2-\beta-\alpha}{\beta-\alpha}\,\frac{T_m(\mu)}{T_{m+1}(\mu)},$$

$$\mu = 1 + 2\,\frac{1-\beta}{\beta-\alpha}, \quad \gamma = \frac{2}{2-\alpha-\beta},$$

并且 $\boldsymbol{y}^0 = \boldsymbol{x}^0$ 和 $\boldsymbol{y}^1 = \boldsymbol{x}^1$。

必须指出的是，精确地估计 α 和 β 对于提高运算速度是很重要的，但要准确估计 α 和 β 常比较困难。

3.7 预处理 Richardson 方法

迭代式(3.2.2)可以写成等价形式

$$\boldsymbol{P}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) = \boldsymbol{r}^k, \quad k \in N,$$

式中：$\boldsymbol{r}^k$ 是在第 k 步的残差；$\boldsymbol{P}$ 必须是非奇异且易于求逆的矩阵。式(3.2.1)中的矩阵 $\boldsymbol{P}$ 可以看做矩阵 $\boldsymbol{A}$ 的预处理器。

预处理过程可以推广成下式：

$$\boldsymbol{P}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) = \alpha \boldsymbol{r}^k, \quad k \in N, \tag{3.7.1}$$

式中：对于每一个 k，$\alpha \neq 0$ 是一个实型加速参数。迭代式(3.7.1)称为预处理 Richardson 方法，对于所有的 k，如果 α 是常数，则称为定常方法；如 α 在迭代过程中变化，则称为非定常方法。

3.7.1 定常预处理 Richardson 方法

定常方法可以按照如下方式递归计算。设 $\boldsymbol{x}^0$ 给定，并且 $\boldsymbol{r}^0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^0$。根据下式依次迭代：

$$\begin{cases} \boldsymbol{P}\boldsymbol{z}^k = \boldsymbol{r}^k, \\ \boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \alpha \boldsymbol{z}^k, \quad k \geqslant 0。\\ \boldsymbol{r}^{k+1} = \boldsymbol{x}^k - \alpha \boldsymbol{A}\boldsymbol{z}^k, \end{cases} \tag{3.7.2}$$

在每一步迭代中，式(3.7.2)需要求解一个矩阵 $\boldsymbol{P}$ 的线性方程组，以及一个涉及到原始矩阵 $\boldsymbol{A}$ 的矩阵——向量积。预处理 Richardson 迭代矩阵如下式：

$$\boldsymbol{R}\alpha = \boldsymbol{I} - \alpha \boldsymbol{P}^{-1}\boldsymbol{A}。\tag{3.7.3}$$

如以 η_i 表示 $\boldsymbol{P}^{-1}\boldsymbol{A}$ 的特征值，则有：

定理 3.7.1　对于任意的非奇异矩阵 $\boldsymbol{P}$，定常预处理 Richardson 方法收敛的充分必要条件是

$$|\eta_i|^2 < \frac{2}{\alpha}\mathrm{Re}\eta_i, \quad \forall i = 1, \cdots, n。\tag{3.7.4}$$

注意到，如果 $\boldsymbol{P}^{-1}\boldsymbol{A}$ 的特征值的实数部分符号不固定，则定常预处理 Richardson 方法不收敛。

推论 3.7.1　如果矩阵 $\boldsymbol{P}^{-1}\boldsymbol{A}$ 具有实数特征值，当且仅当

$$1 < \frac{2}{\alpha\eta_i}, \quad \forall i = 1, \cdots, n, \tag{3.7.5}$$

则定常预处理 Richardson 方法收敛。

如果从此假设 $\boldsymbol{P}^{-1}\boldsymbol{A}$ 是对称并正定的，则条件(3.7.5)减弱为

$$0 < \alpha < \frac{2}{\eta_{\max}},$$

式中：$\eta_{\max}$是 $\boldsymbol{P}^{-1}\boldsymbol{A}$ 的最大特征值。特别地，以$\{\boldsymbol{v}_i \mid i=1,\cdots,n\}$表示由 $\boldsymbol{P}^{-1}\boldsymbol{A}$ 的特征向量构成的正交基，据此将误差 $\boldsymbol{e}^k$ 展开成

$$\boldsymbol{e}^k = \sum_{i=1}^{n} e_i^k \boldsymbol{v}_i, \quad k \in N。$$

从式(3.2.4)，可以推导出

$$e_i^{k+1} = (1-\alpha\eta_i)e_i^k, \quad i=1,\cdots,n; k \in N。 \tag{3.7.6}$$

实数 $\rho_i = |1-\alpha\eta_i|$，称为缩减因子(见图 3.1)。很明显，当且仅当 $\rho_i < 1, i=1,\cdots,n$ 时，可以达到收敛，因为 R_α 的谱半径是$\max\limits_{1 \leqslant i \leqslant n} \rho_i$。

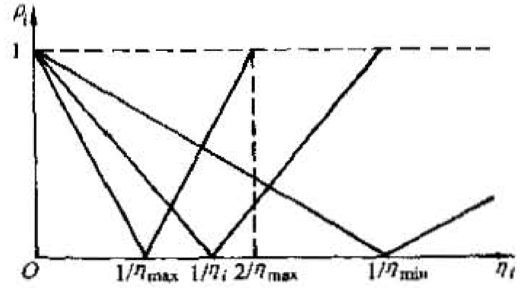


图 3.1 定常预处理 Richardson 方法的缩减因子

α 的最优值 α^* 满足

$$|1-\alpha^* \eta_{\min}| = |1-\alpha^* \eta_{\max}|,$$

式中：$\eta_{\min}$和 $\eta_{\max}$分别表示 $\boldsymbol{P}^{-1}\boldsymbol{A}$ 的最小和最大的特征值。因此

$$\alpha^* = \frac{2}{\eta_{\min} + \eta_{\max}}。 \tag{3.7.7}$$

相应地，可得

$$\rho^* = \rho(R_{\alpha^*}) = 1 - \alpha^* \eta_{\min} = \frac{\eta_{\max} - \eta_{\min}}{\eta_{\max} + \eta_{\min}}。$$

如果 $\boldsymbol{P}^{-1}\boldsymbol{A}$ 是对称正定矩阵，可以得到

$$\chi_{sp}(\boldsymbol{P}^{-1}\boldsymbol{A}) = \frac{\eta_{\max}}{\eta_{\min}}, \tag{3.7.8}$$

由此，可以推导出

$$|\boldsymbol{e}^{k+1}| \leqslant \rho^* |\boldsymbol{e}^k|, \quad \forall k \in N, \tag{3.7.9}$$

这里：

$$\rho^* = \frac{\chi_{sp}(\boldsymbol{P}^{-1}\boldsymbol{A}) - 1}{\chi_{sp}(\boldsymbol{P}^{-1}\boldsymbol{A}) + 1}。 \tag{3.7.10}$$

因子 ρ^* 称为带有优化参数 α^* 之定常预处理 Richardson 方法的缩减因子。$\chi_{sp}(\boldsymbol{P}^{-1}\boldsymbol{A})$越小，则收敛越快。

3.7.2 预处理 Richardson 最小残差法

必须指出的是，如果 $\boldsymbol{P}^{-1}\boldsymbol{A}$ 不满足对称正定矩阵的条件(这种情况是很常见的)，但矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{A}$ 都是对称并且正定的，则前面的结论仍然可用。事实上，$\boldsymbol{P}^{-1}\boldsymbol{A}$ 仍然具有正实数特征值，

因为关于标量积$(\cdot,\cdot)_P$可以证明$\boldsymbol{P}^{-1}\boldsymbol{A}$仍然是对称且正定的，标量积的定义如下：

$$(\boldsymbol{u},\boldsymbol{v})_P=(\boldsymbol{P}\boldsymbol{u},\boldsymbol{v}),\quad |\boldsymbol{u}|_P=\sqrt{(\boldsymbol{u},\boldsymbol{v})_P}\quad \forall\, \boldsymbol{u},\boldsymbol{v}\in\Re^n。\tag{3.7.11}$$

上面的$(\cdot,\cdot)_P$表示欧几里德标量积。很明显，$(\cdot,\cdot)_P$是$\Re^n$空间中的标量积，并且$(\cdot,\cdot)_P$是相关的向量的范数。因此，得到下列结果：

引理 3.7.1　假设$\boldsymbol{P}$是对称正定矩阵。如果$\boldsymbol{A}$是对称的（正定的），则$\boldsymbol{P}^{-1}\boldsymbol{A}$是关于由式(3.7.11)所定义标量积$(\cdot,\cdot)_P$对称的（正定的）。进一步，如果$\boldsymbol{A}$是对称的，则谱条件数$\chi_{sp}(\boldsymbol{P}^{-1}\boldsymbol{A})$就和基于自然矩阵范数$\|\cdot\|_P$（由向量范数$|\cdot|_P$所引入）的$\boldsymbol{P}^{-1}\boldsymbol{A}$的条件数相等。

现在假定$\boldsymbol{A}$是正定矩阵，而$\boldsymbol{P}$既是正定的又是对称的。由式(3.7.2)，可得

$$\boldsymbol{z}^{k+1}=\boldsymbol{z}^{k+1}(\alpha)=\boldsymbol{P}^{-1}(\boldsymbol{r}^k-\alpha\boldsymbol{A}\boldsymbol{z}^k),$$

因此

$$|\boldsymbol{z}^{k+1}|_P^2=|\boldsymbol{P}^{-1}\boldsymbol{r}^k|_P^2+\alpha^2|\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{z}^k|_P^2-2\alpha(\boldsymbol{P}^{-1}\boldsymbol{r}^k,\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{z}^k)_P。$$

现在，通过选择α的值，就是α_k，使$|\boldsymbol{z}^{k+1}(\alpha)|_P^2$最小，也就是

$$\alpha_k\text{ 是满足方程}\left(\frac{\mathrm{d}}{\mathrm{d}\alpha}|\boldsymbol{z}^{k+1}(\alpha)|_P^2\right)_{|\alpha=\alpha_k}=0\text{ 的值。}\tag{3.7.12}$$

通过简单的计算，得到

$$\alpha_k=\frac{(\boldsymbol{z}^k,\boldsymbol{A}\boldsymbol{z}^k)}{(\boldsymbol{A}\boldsymbol{z}^k,\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{z}^k)}。\tag{3.7.13}$$

除了$\boldsymbol{z}^k=0$以外，分子总是保持正值。$\boldsymbol{z}^k=0$的情况下有$\boldsymbol{r}^k=0$，这时有$\boldsymbol{x}^k=\boldsymbol{x}$（确切解）。

式(3.7.13)给出了以α_k代替α后式(3.7.2)的迭代方法，称为预处理 Richardson 最小残差法(PR-MR)。事实上，如果$\boldsymbol{P}$是单位矩阵，则有$\boldsymbol{z}^{k+1}=\boldsymbol{r}^{k+1}$，并且$\alpha_k$将最小化新的残差向量$\boldsymbol{r}^{k+1}$的欧几里德范数$|\boldsymbol{r}^{k+1}|$。

3.7.3　预处理 Richardson 最速下降法

假设$\boldsymbol{A}$是对称且正定的，则也可以定义预处理 Richardson 最速下降法(PR-SD)。从式(3.7.2)，可以推导出

$$\boldsymbol{r}^k=\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}^k=\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}^k)=-\boldsymbol{A}\boldsymbol{e}^k,$$

这里$\boldsymbol{e}^k$是在式(3.2.3)中定义的误差项。根据式(3.2.4)和(3.7.3)，可以得到

$$\boldsymbol{e}^{k+1}=\boldsymbol{e}^{k+1}(\alpha)=(\boldsymbol{I}-\alpha\boldsymbol{P}^{-1}\boldsymbol{A})\boldsymbol{e}^k,\tag{3.7.14}$$

因此（通过标量积$(\boldsymbol{u},\boldsymbol{v})_A=(\boldsymbol{A}\boldsymbol{u},\boldsymbol{v})$引入的$|\cdot|_A$来定义向量范数）

$$\begin{aligned}|\boldsymbol{e}^{k+1}|_A^2&=(\boldsymbol{A}\boldsymbol{e}^{k+1},\boldsymbol{e}^{k+1})\\&=-(\boldsymbol{r}^{k+1},\boldsymbol{e}^{k+1})\\&=-(\boldsymbol{r}^k,\boldsymbol{e}^k)+\alpha[(\boldsymbol{r}^k,\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{e}^k)+(\boldsymbol{A}\boldsymbol{z}^k,\boldsymbol{e}^k)]-\alpha^2(\boldsymbol{A}\boldsymbol{z}^k,\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{e}^k),\end{aligned}$$

这样，将$\alpha=\alpha_k$代入式(3.7.2)就可以得到 PR-SD 迭代法，这里

$$\alpha_k\text{ 是满足方程}\left(\frac{\mathrm{d}}{\mathrm{d}\alpha}|\boldsymbol{e}^{k+1}(\alpha)|_A^2\right)_{|\alpha=\alpha_k}=0\text{ 的值。}\tag{3.7.15}$$

将$\boldsymbol{x}^{k+1}(\alpha)$展开成$\boldsymbol{x}^{k+1}(\alpha)=\boldsymbol{x}^k+\alpha\boldsymbol{z}^k$，这等价为最小化$\phi(\boldsymbol{x}^{k+1}(\alpha))$，这里的$\phi(\boldsymbol{w})$是二次泛函，

$$\phi(\boldsymbol{w})=\frac{1}{2}(\boldsymbol{A}\boldsymbol{w},\boldsymbol{w})-(\boldsymbol{b},\boldsymbol{w})。$$

通过简单的计算，可以得到

$$\alpha_k = \frac{1}{2}\frac{(r^k, P^{-1}Ae^k) + (Az^k, e^k)}{(Az^k, P^{-1}Ae^k)} = \frac{1}{2}\frac{(r^k, z^k) + (Az^k, A^{-1}r^k)}{(Az^k, z^k)},$$

因此

$$\alpha_k = \frac{(z^k, r^k)}{(z^k, Az^k)}。\tag{3.7.16}$$

同样，除了 $r^k=0$ 以外，上式的分子不等于 0。

注意到有

$$\frac{|e^k|_A^2 - |e^{k+1}|_A^2}{|e^k|_A^2} = \frac{(z^k, z^k)_P^2}{(P^{-1}Az^k, z^k)_P(A^{-1}Pz^k, z^k)_P}。$$

再次使用 Kantorovich 不等式，可以得到下面的误差估计：

$$|e^{k+1}|_A \leqslant \rho^* \ |e^k|_A, \quad k \in N,\tag{3.7.17}$$

上式中的缩减因子 ρ^* 由式(3.7.10)给出。

上述迭代方法的收敛渐近速率的一种定义式如下：

$$\sigma = -\log\rho^*,\tag{3.7.18}$$

有时，可以用另外一个概念来表达某种方法的收敛特性，这种概念基于对 k 步的残差缩减因子平均的基础上。

上面 σ 的倒数，记为 I，用于度量将初始误差递减 1/e 所需要的平均迭代次数。收敛速度越大，则求解到给定精度所需要的迭代次数越少。对于上面所讲述的预处理 Richardson 方法，相当于式(3.7.7)、式(3.7.13)和式(3.7.16)的所有情况，必需的迭代次数按下式递增：

$$I \approx \frac{1}{2}\chi_{sp}(P^{-1}A)。\tag{3.7.19}$$

因此，选择矩阵 P 可以认为是一种减小 A 的条件数和计算迭代次数的手段。在这方面，P 可以认为是 A 的预处理器，并且在本节中所介绍的方法都是预处理 Richardson 方法。对 P 是单位矩阵的情况，也可以认为是 Richardson 方法。

如果 P 是单位矩阵，则 $z^k=r^k$。这样，式(3.7.16)变为

$$\alpha^k = \frac{|r^k|^2}{(r^k, Ar^k)},$$

并且由式(3.7.2)可以推导出

$$x^{k+1} = x^k + \frac{|r^k|^2}{(r^k, Ar^k)}r^k。\tag{3.7.20}$$

这就是梯度方法或者称为 Richardson 最速下降法，Cauchy 对此已经有所论述。如果 P 不是单位矩阵，则按照式(3.7.16)选择 α_k 的迭代方案(3.7.2)也称为预处理梯度法。

因为迭代式(3.2.2)等价为

$$P(x^{k+1} - x^k) = r^k, \quad k \in N。$$

可以将 Jacobi 方法、Guass-Seidel 方法、S. O. R. 以及 S. S. O. R. 等方程求解方法看作是加速参数 $\alpha=1$ 并且预处理器 P 分别由式(3.3.2)、(3.4.1)、(3.5.2)以及(3.5.7)给出的预处理 Richardson 方法。

3.8　共轭梯度法

假设 A 是对称正定矩阵，容易证明求解式(3.1.1)等价于将二次泛函

$$\phi(w)=\frac{1}{2}(Aw,w)-(b,w) \tag{3.8.1}$$

最小化。事实上，ϕ 的最小值就是 $-(1/2)(A^{-1}b,b)$，并且可以得到 $x=A^{-1}b$。注意到

$$r^k=-\nabla\phi(x^k), \tag{3.8.2}$$

这样，式(3.3.3)的残差就是二次泛函 ϕ 在 x^k 处的负梯度。

当 $P=I$，Richardson 方法通过沿着 ϕ 的梯度方向迭代更新：

$$x^{k+1}=x^k+\alpha_k r^k。 \tag{3.8.3}$$

在共轭梯度法中(CG)，迭代式的更新仍为式(3.8.3)，不过用一个与梯度方向并不平行的新方向 p^k 替换式(3.8.3)中的 r^k。共轭梯度法最初是由 Hestenes 和 Stiefel 提出的，用以求解对称正定线性系统，因为这种方法(在没有舍入误差的情况下)可以用很少步骤得到方程的精确解。

在共轭梯度法中，方向$\{p^k\}$是 A 类共轭，即它们满足正交性：对于 $m\neq j$，有$(p^j,Ap^m)=0$。特别有

$$(p^{k+1},Ap^k)=0,\quad \forall k\in N。 \tag{3.8.4}$$

共轭梯度法的思路为，设 $p^0,\cdots,p^m$ 是线性独立向量，x^0 是解的初始猜测，构造如下序列：

$$x^{k+1}=x^k+\alpha_k p^k,\quad 0\leqslant k\leqslant m,$$

式中的 α_k 是非零实数。当且仅当 p^j 是 A 类共轭并且 $\alpha^k=\dfrac{(r^k,p^k)}{(p^k,Ap^k)}$时，则 x^{k+1} 在$(k+1)$维超平面

$$w=x^0+\sum_{j=0}^{k}\gamma_j p^j,\quad \gamma_j\in\Re$$

上最小化函数 ϕ。因此，共轭梯度法可以表达如下。设 x^0 是解的初始猜测，$r^0=b-Ax^0$，并且设 $p^0=r^0$。对于每一个 $k\in N$，迭代式由如下公式给出：

$$\alpha^k=\frac{(r^k,p^k)}{(p^k,Ap^k)}=\frac{|r^k|^2}{(p^k,Ap^k)}, \tag{3.8.5}$$

$$x^{k+1}=x^k+\alpha_k p^k, \tag{3.8.6}$$

$$r^{k+1}=r^k-\alpha_k Ap^k, \tag{3.8.7}$$

$$\beta_{k+1}=\frac{(r^{k+1},Ap^k)}{(p^k,Ap^k)}=\frac{|r^{k+1}|^2}{|r^k|^2}, \tag{3.8.8}$$

$$p^{k+1}=r^{k+1}+\beta_{k+1}p^k。 \tag{3.8.9}$$

要证明系数 α_k 和 β_{k+1} 可以写成如上面所示的二阶形式，首先需要确认存在

$$(p^k,r^{k+1})=0,\quad (p^k,Ap^k)=(r^k,Ap^k),\quad (r^{k+1},r^k)=0。 \tag{3.8.10}$$

注意到上面的求解过程式(3.8.5)～(3.8.9)要求 $p^k\neq 0$。不过，即使 $p^k=0$ 的情况下，也有

$$r^k=-\beta_k p^{k-1},\quad k\geqslant 1,$$

并且有

$$|r^k|^2=-\beta_k(p^{k-1},r^k)=0,$$

因此 $x^k=A^{-1}b$ 就是所需要的解。

另一方面，当且仅当 $p^k=0$ 或 $\alpha_k=0$，迭代项 x^{k+1} 才等于前一个迭代项 x^k，这样再次证明 $x^k=A^{-1}b$。对于任意固定的方向矢量 p^k，设

$$e^{k+1}(\alpha)=x^{k+1}(\alpha)-x=(x^k+\alpha p^k)-x,$$

式(3.8.5)中给出的 α_k 值满足

$$\left(\frac{\mathrm{d}}{\mathrm{d}\alpha}\,|\,e^{k+1}(\alpha)\,|_A^2\right)_{|\alpha=\alpha_k}=0, \tag{3.8.11}$$

因此,α_k 是所有 $\alpha\in\Re$ 中,使函数 $\phi(x^k+\alpha p^k)$ 最小化的最优参数。另一方面,式(3.8.8)中给出的值 β_{k+1} 保证式(3.8.4)成立。

共轭梯度法迭代收敛特征有下面的误差估计式:

$$|\,e^k\,|_A\leqslant 2\left[\frac{\sqrt{\chi_{sp}(\boldsymbol{A})}-1}{\sqrt{\chi_{sp}(\boldsymbol{A})}+1}\right]^k|\,e^0\,|_A,\quad k\in N。 \tag{3.8.12}$$

如果仍然用 $\boldsymbol{p}$ 代表预处理器,$\boldsymbol{p}$ 是对称正定矩阵,则对于 $k>0$ 的预处理共轭梯度法(PCG)可表示如下:

$$\alpha_k=\frac{(z^k,r^k)}{(p^k,Ap^k)}, \tag{3.8.13}$$

$$x^{k+1}=x^k+\alpha_k p^k, \tag{3.8.14}$$

$$r^{k+1}=r^k-\alpha_k Ap^k, \tag{3.8.15}$$

$$Pz^{k+1}=r^{k+1}, \tag{3.8.16}$$

$$\beta_{k+1}=\frac{(z^{k+1},r^{k+1})}{(z^k,r^k)}, \tag{3.8.17}$$

$$p^{k+1}=z^{k+1}+\beta_{k+1}p^k。 \tag{3.8.18}$$

对于上面的情况,可以像共轭梯度法一样来完成初始化过程,取一个初始解 x^0,并设置 $r^0=b-Ax^0$,$p^0=z^0=P^{-1}r^0$。误差估计式(3.8.12)成为

$$|\,e^k\,|_A\leqslant 2\left[\frac{\sqrt{\chi_{sp}(\boldsymbol{P}^{-1}\boldsymbol{A})}-1}{\sqrt{\chi_{sp}(\boldsymbol{P}^{-1}\boldsymbol{A})}+1}\right]^k|\,e^0\,|_A,\quad k\in N。 \tag{3.8.19}$$

如果选择适当的 $\boldsymbol{P}$,缩减因子可以大幅度小于共轭梯度法中的缩减因子。另一方面,这里强调在预处理共轭梯度法中的每一步,都需要求解线性系统(3.8.16)。

这样,可以得到一个共轭梯度迭代方法常用的停止准则如下:

$$|\,r^k\,|\leqslant\varepsilon\,|\,b\,|。 \tag{3.8.20}$$

上式中的 $\varepsilon>0$ 是指定的容许误差。

表 3.1 总结了上述方法的基本特点。

表 3.1　迭代方法总结

	A	α_k	最小化	误差估计
静态 Richardson 法	$A=A^T>0$	$\frac{2}{\eta_{min}+\eta_{max}}$	$\rho(I-\alpha P^{-1}A)$	$\|e^k\|_P\leqslant\left(\frac{c-1}{c+1}\right)^k\|e^0\|_P$
残差最小化 Richardson 法	$A>0$	$\frac{(z^k,Az^k)}{(Az^k,P^{-1}Az^k)}$	$\|z^{k+1}\|_P=\|e^{k+1}\|_{A^TP^{-1}A}$	如果 $A=A^T$ $\|z^k\|_P\leqslant\left(\frac{c-1}{c+1}\right)^k\|z^0\|_P$
Richardson 最速下降法(梯度法)	$A=A^T>0$	$\frac{(z^k,r^k)}{(z^k,Az^k)}$	$\|e^{k+1}\|_A$ 或者 $\phi(x^{k+1})$	$\|e^k\|_A\leqslant\left(\frac{c-1}{c+1}\right)^k\|e^0\|_A$
共轭梯度法	$A=A^T>0$	$\frac{(z^k,r^k)}{(p^k,Ap^k)}$	$\|e^{k+1}\|_A$ 或者 $\phi(x^{k+1})$	$\|e^k\|_A\leqslant 2\left(\frac{\sqrt{c}-1}{\sqrt{c}+1}\right)^k\|e^0\|_A$

3.9　多网格方法

当迭代矩阵 $\boldsymbol{B}$ 的谱半径接近 1 时，迭代收敛速度较慢。傅里叶分析表明：是迭代误差的低频部分造成矩阵 $\boldsymbol{B}$ 的特征值较大，抑制了收敛速度，而高频误差对迭代收敛速度的影响较小。也就是说图像上大范围的误差对收敛速度的影响较大，而局部或细部误差对收敛速度的影响较小。

多网格方法的基本思想是，在偏微分方程离散化成为差分方程的过程中采用不同大小的网格。事实上，对宽度为 h 的网格上的迭代误差进行平滑处理后，在宽度为 $2h$ 的粗网格上进行迭代计算，则误差收敛速度会加快；同时，在宽度为 $2h$ 的网格上无法处理的高频误差，却可以在更精细的网格中进行迭代计算而得到消除。图像处理中常采用小波变换，对图像进行逐级平滑。

多网格方法可以表示成如下形式：

$$A_j \boldsymbol{x}_{(j)} = \boldsymbol{b}_{(j)}, \quad j = 0, \cdots, m, \tag{3.9.1}$$

式中：A_j 是在一个有限维空间 V^j 中的可逆线性算子，有 $\dim V^j < \dim V^{j+1}, j=0,\cdots,m-1$；$\boldsymbol{b}_{(j)}$ 是等式右边给定值。

为求解在 $j=m$ 时的式(3.9.1)，可以在最高层省略下标，即

$$V = V^m, \quad A = A_m, \quad \boldsymbol{b} = \boldsymbol{b}_{(m)}, \quad \boldsymbol{x} = \boldsymbol{x}_{(m)},$$

算子 $A_j(j<m)$ 将在多网格方法中起辅助作用。

对于每一个 j，可以认为式(3.9.1)是差分方程在宽度为 h_j 的网格上的离散化表达式。假设 $h_{j-1}=2h_j$，则式(3.9.1)代表了一系列带有细化因子 2 的离散化过程。

为求解问题(3.9.1)，考虑迭代方法

$$\boldsymbol{x}_{(j)}^{k+1} = \boldsymbol{x}_{(j)}^{k} + P_j(\boldsymbol{b}_{(j)} - A_j \boldsymbol{x}_{(j)}^{k}), \tag{3.9.2}$$

式中：P_j 是 A_j 的预处理器。与其相关的误差 $\boldsymbol{e}_{(j)}^{k}$ 根据下式转换：

$$\boldsymbol{e}_{(j)}^{k+1} = B_j \boldsymbol{e}_{(j)}^{k}, \quad B_j = I - P_j A_j。 \tag{3.9.3}$$

多网格循环点算法可以看作是式(3.9.2)在 $j=m$ 时的加速算法，其计算步骤如下：

(1) 在精细网格的预平滑

对 $\boldsymbol{x}_{(j)}^{0}$，进行 n_1 次计算有

$$\boldsymbol{x}_{(j)}^{k} = \boldsymbol{x}_{(j)}^{k-1} + P_j(\boldsymbol{b}_{(j)} - A_j \boldsymbol{x}_{(j)}^{k-1}), \quad k = 1, \cdots, n_1。 \tag{3.9.4}$$

(2) 粗网格校正

① 计算残差或误差

$$\boldsymbol{r}_{(j)} = \boldsymbol{b}_{(j)} - A_j \boldsymbol{x}_{(j)}^{n_1}.$$

② 在粗网格对残差进行变换，设

$$\boldsymbol{r}_{(j-1)} = R_j^{j-1} \boldsymbol{r}_{(j)},$$

这里

$$R_j^{j-1} : V^j \to V^{j-1}$$

是平滑变换算子。

③ 求解粗网格方程

$$A_{j-1} \boldsymbol{x}_{(j-1)} = \boldsymbol{r}_{(j-1)}。 \tag{3.9.5}$$

④ 校正在 V^j 空间的解，设

$$\bar{x}_{(j)} = x_{(j)}^{n_1} + \Pi_{j-1}^{j} x_{(j-1)},$$

这里

$$\Pi_{j-1}^{j} : V^{j-1} \to V^j$$

是插值反变换算子。

(3) 精细网格下的后平滑处理

设 $\hat{x}_{(j)}^0 = \bar{x}_{(j)}$，进行 n_2 次计算，

$$\hat{x}_{(j)}^k = \hat{x}_{(j)}^{k-1} + P_j(b_{(j)} - A_j\hat{x}_{(j)}^{k-1}), \quad k = 1, \cdots, n_2。 \tag{3.9.6}$$

插值算子和平滑算子都是满秩线性映射。预平滑或后平滑步数 n_1 或 n_2 中可以有一个为 0。大多数情况下，预平滑或后平滑步数 n_1 或 n_2 可根据 j 的大小决定。

考虑粗网格问题(3.9.5)，如果 V^{j-1} 的维数较低，则可以利用直接方法精确求解。这样，就可以采用两种宽度不同的网格的循环算法。如果粗网格方程(3.9.5)不能得到确切解，则可以在多种不同宽度的网格上循环计算求解。

参 考 文 献

[1] Quarteroni A, Valli A. Numerical Approximation of Partial Differential Equations. New York: Springer-Verlag, 1997

[2] Isaacson E, Keller H B. Analysis of Numerical Methods. New York: Dover Publication Inc., 1994

[3] Horn R A, Johnson C R. Matrix Analysis. Cambridge: Cambridge University Press, 1985

[4] Golub G H, Loan C F. Matrix Computations. Baltimore: The Johns Hopkins University Press, 1989

[5] Young D M. Iterative Solution of Large Linear Systems. New York: Academic Press, 1971

[6] 蔡大用. 数值代数. 北京:清华大学出版社,1987

[7] 曹志浩,张玉德,李瑞遐. 矩阵计算和方程求根. 北京:高等教育出版社,1979

[8] 黄友谦. 数值实验. 北京:高等教育出版社,1989

[9] 黄家赣. 线性代数方程组的迭代解法. 北京:科学出版社,1997

[10] 蒋尔雄. 对称矩阵计算. 上海:上海科学技术出版社,1984

[11] 王德人,杨忠华. 数值逼近引论. 北京:高等教育出版社,1990

[12] 徐树方. 矩阵计算的理论与方法. 北京:北京大学出版社,1999

第4章 轮廓线匹配

轮廓线匹配是计算机视觉研究中的一个重要问题，应用非常广泛。例如：在医学图像处理中，要根据三维断层图上器官的轮廓线来重建整个器官的三维表面形状，或从一组心脏图形中找出左心室壁运动的情况[1]；在机器人视觉中，需要找到轮廓相同的零件[2]。传统的轮廓线匹配方法大多是基于轮廓线特征或凹/凸结构的形式符号法描述，其思想是在匹配前将轮廓线简化成与形状有关的链码，然后用链码匹配技术去寻找相匹配的子链码。虽然用这些方法对于比较简单的轮廓线匹配能节约计算时间，但匹配的精度不高，对于比较复杂的轮廓线则很难得到可靠的匹配[3,4]。

本章介绍了用曲率来描述轮廓的形状进行轮廓匹配的一种数值算法。这是在两幅时间或空间位置不同的图像中，通过优化一个二项指标函数来完成轮廓匹配。其中第一项为弯曲变形指标，反映轮廓变形量；第二项是弹性约束项，用于表示相邻点弹性约束的状况[5]。

本文用曲率作为轮廓线形状的特征描述，因为曲率有以下性质：

(1) 轮廓线经过旋转，或平移后，其曲率是不变的。

(2) 两条轮廓线上相应点的曲率变化可以解释成弯曲变形，即可以用力学上的弯曲变形量来度量两条轮廓线的差别[1]。

(3) 用曲率描述轮廓线的形状，能将二维平面上的轮廓线用一维曲率变化来描述，这样能减少计算量。

本章是这样安排的：首先第4.1节介绍了曲线和曲率的一些基本概念。第4.2节导出了关于轮廓线匹配的弹性约束项，由此得到的最优匹配方程物理意义明确，比较容易解。第4.3节介绍了如何用有限差迭代法解这个方程。第4.4节是对结构复杂的轮廓线采用多尺度法计算的讨论。第4.5节用实验对算法作了验证。

4.1 平面微分几何初步

4.1.1 曲线的定义

平面上的曲线或轮廓线可以定义为线到面的映射，即：映射 $C(p):[a,b]\in\Re\rightarrow\Re^2$ 定义了一条平面曲线，其中 p 是曲线的参数。对于每个值 $p_0\in[a,b]$，可以得到平面上一个点 $C(p_0)=[x(p_0),y(p_0)]$。为了讨论方便，假设曲线是二次可微的。

如果一条曲线有 $C[a]=C[b]$，则该曲线是闭曲线。如果一条曲线上至少有一对参数 $p_0\neq p_1$，$p_0,p_1\in(a,b)$，使 $C(p_0)=C(p_1)$，那么该曲线有自交叉点。没有自交叉点的曲线称为简单曲线。

参数 p 可以有很多选择。例如，参数 p 可以看作是一个动点在曲线上运动的速度，这个速度可以用曲线上切线矢量 $\frac{\partial C}{\partial p}$ 表示。也可以将曲线的弧长作为参数 p，对于曲线的弧长有

$$\left\|\frac{\partial C}{\partial p}\right\| = 1,$$

式中：$\|\cdot\| = \sqrt{\langle\cdot,\cdot\rangle}$是欧氏空间中的长度。弧长的定义区间不再是$[a,b]$，而是区间$[0,L]$，其中 L 是曲线的长度。曲线长度是唯一的，可以根据公式

$$\frac{\partial C}{\partial s} = \frac{\mathrm{d}C}{\mathrm{d}p}\frac{\mathrm{d}p}{\mathrm{d}s},$$

$$\frac{\mathrm{d}s}{\mathrm{d}p} = \left[\left(\frac{\mathrm{d}x}{\mathrm{d}p}\right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}p}\right)^2\right]^{\frac{1}{2}}$$

计算。

本文中讨论的曲线在任意两点之间其长度是有限的，相当于认为函数 $x(p)$和 $y(p)$的变分是有界的。根据弧长的定义，两点间弧长的计算公式为

$$L(p_0,p_1) = \int_{p_0}^{p_1}\left[\left(\frac{\mathrm{d}x}{\mathrm{d}p}\right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}p}\right)^2\right]^{\frac{1}{2}}\mathrm{d}p = \int_{s(p_0)}^{s(p_1)}\mathrm{d}s。$$

4.1.2 曲线的参数表示

在直角坐标系中，曲线 $C(s)$可以用 $x(s)$和 $y(s)$表示，$0<s<L$，其中，s 是曲线的弧长，而 L 是曲线的总长度。为了对数字图像中的曲线进行数值计算，必须求曲线在光栅网格上的离散表示。这会有曲线的非均匀采样问题。为了得到均匀采样，以保证计算的准确性，需要用插值法对曲线进行再次采样。在计算中，可以采用三点 Lagrange 插值法。

4.1.3 曲率的定义

曲率 k 的定义为

$$k = \left\|\frac{\partial^2 C}{\partial s^2}\right\|。$$

如果 $\boldsymbol{T}$ 和 $\boldsymbol{N}$ 分别代表单位切向量和法向量($\boldsymbol{T}\perp\boldsymbol{N}$)，那么

$$\frac{\mathrm{d}C}{\mathrm{d}s} = \boldsymbol{T},$$

$$\frac{\mathrm{d}^2 C}{\mathrm{d}s^2} = k\boldsymbol{N}。$$

因此，可得 Frenet 公式

$$\frac{\mathrm{d}\boldsymbol{T}}{\mathrm{d}s} = k\boldsymbol{N}, \qquad \frac{\mathrm{d}\boldsymbol{N}}{\mathrm{d}s} = -k\boldsymbol{T}。$$

曲率的特性：

(1) 只有两种曲线的曲率为常数：直线(曲率为零)和圆(曲率为半径的倒数)。曲率为常数的闭曲线只能是圆。

(2) 在曲线上的顶点处，曲率不存在一阶导数。

(3) 闭曲线的曲率总和是 2π 的倍数(简单曲线曲率总和为 2π)。

(4) 所有长度为 L 的简单闭曲线中，半径为 $L/2\pi$ 的圆的面积最大。

4.1.4 曲率计算

上述已介绍过，曲线可以用 $C(s)=[x(s),y(s)]$表示。并且假设，曲线是二次可微的，可

以定义

$$\dot{x}(s)=\frac{\mathrm{d}x}{\mathrm{d}s},\qquad \ddot{x}(s)=\frac{\mathrm{d}^2x}{\mathrm{d}s^2}。$$

仿此，有 $\dot{y}(s)$ 和 $\ddot{y}(s)$。

曲率可根据下列公式计算：

$$k(s)=\frac{\dot{x}(s)\ddot{y}(s)-\ddot{x}(s)\dot{y}(s)}{[\dot{x}(s)^2+\dot{y}(s)^2]^{\frac{3}{2}}}。\tag{4.1.1}$$

4.1.5　高斯滤波器

离散化的曲线不可避免地带有噪声。曲线上的微小变化都会由于微分而放大，使曲率计算产生严重误差。这就要求离散化的曲线在微分以前必须进行某种程度的平滑。根据 Marr 提出的方法，将 $x(s)$ 与高斯函数的一阶和二阶导数进行卷积得到 $\dot{x}(s)$ 和 $\ddot{x}(s)$。$\dot{y}(s)$ 和 $\ddot{y}(s)$ 也可使用同样方法得到。选择高斯滤波器是因为它对于平面曲线有多尺度特性。高斯滤波器的另一个重要优点是当平滑程度增加时，轮廓线不会产生新的过零点[6]。

用 * 表示卷积算子，定义：

$$\dot{X}(s,\sigma)=x(s)*\dot{g}(s,\sigma),\qquad \ddot{X}(s,\sigma)=x(s)*\ddot{g}(s,\sigma)。$$

同样有 $\dot{Y}(s,\sigma)$ 和 $\ddot{Y}(s,\sigma)$。式中：

$$g(s,\sigma)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{s^2}{2\sigma^2}\right),\tag{4.1.2}$$

$$\dot{g}(s,\sigma)=\frac{\partial}{\partial s}g(s,\sigma)=\frac{-s}{\sigma^3\sqrt{2\pi}}\exp\left(-\frac{s^2}{2\sigma^2}\right),\tag{4.1.3}$$

$$\ddot{g}(s,\sigma)=\frac{\partial^2}{\partial s^2}g(s,\sigma)=\left(\frac{s^2}{\sigma^5\sqrt{2\pi}}-\frac{1}{\sigma^3\sqrt{2\pi}}\right)\exp\left(-\frac{s^2}{2\sigma^2}\right),\tag{4.1.4}$$

分别是高斯函数及其一阶导数和二阶导数。

这样，式(4.1.1)可以重新写成具有高斯尺度参数 σ 的曲率计算公式：

$$k(s,\sigma)=\frac{\dot{X}(s,\sigma)\ddot{Y}(s,\sigma)-\ddot{X}(s,\sigma)\dot{Y}(s,\sigma)}{[\dot{X}^2(s,\sigma)+\dot{Y}^2(s,\sigma)]^{\frac{3}{2}}}。\tag{4.1.5}$$

4.2.　数学模型

4.2.1　问题的背景

设 C_P，C_Q 为一个图像序列中两个连续、光滑的轮廓线；C_Q 由 C_P 弹性变形而得。轮廓 C_Q 和 C_P 分别是参数 s 和 s' 的函数，其中 s 和 s' 是沿轮廓的弧长。轮廓 C_P 用参数表达式 $x_P(s)$ 和 $y_P(s)$ 表示；同样，对轮廓 C_Q 有 $x_P(s')$ 和 $y_P(s')$。s 和 s' 的起点应选在曲率比较小的区域内，以便减小起点误差对指标函数的影响。

现在，需对 C_P 上的每个点 s，确定 C_Q 上的对应点 s'。为此，需定义一个变形量，它能比较

s 和 s' 领域的局部变形量。假设曲率 k_P 和 k_Q 在这些领域内基本保持不变，可引入以下形式的局部变形指标：

$$E_{\text{curve}} = \int_{\delta s} (k_Q(s') - k_P(s))^2 \mathrm{d}s, \tag{4.2.1}$$

其中：δs 是 C_P 上的一小段。

同时，希望变形量沿轮廓平缓变化。Duncan[1] 使用了一个弹性约束项。于是，指标函数可以按如下形式表示：

$$E = E_{\text{curve}} + \lambda E_{\text{elastic}}, \tag{4.2.2}$$

其中：

$$E_{\text{elastic}} = \oint_{C_P} \left\| \frac{\partial(Q(s') - P(s))}{\partial s} \right\| \mathrm{d}s。 \tag{4.2.3}$$

E_{elastic} 描述变形量 P,Q 沿 C_P 的变化，$\|\cdot\|$ 记为在空间 $\Re^2$ 上以欧几里德内积 $\langle\cdot,\cdot\rangle$ 定义的范数。λ 是相应的加权因子（λ 值大表示弹性约束程度大）。Duncan 提出的方法是，先将式(4.2.2)离散化，然后用优化方法求解。然而该方法可能造成两条轮廓线上点不是按顺序一一对应。为了解决这个问题，Cohen 提出下列数学模型[5]：

给定两条轮廓线 C_Q 和 C_P，分别以参数 s 和 s' 表示，$s \in [0,1]$，$s' \in [0,\alpha]$（α 为曲线 C_Q 的总长度），要确定一个函数 f：$[0,1] \to [0,\alpha]$ 满足：

$$f(0) = 0 \text{ 和 } f(1) = \alpha \tag{4.2.4}$$

和

$$f = \operatorname{argmin} E(f), \tag{4.2.5}$$

其中：E 在式(4.2.2)中定义。

Cohen 通过变分法来求 f。它根据 Euler-Lagrange 方程求方程 $\nabla E(f) = 0$ 的解，即函数 $E(f)$ 的一个局部极小值：

$$f'' \|Q'(f)\|^2 + k_P \langle N_P, Q'(f) \rangle + \frac{1}{\lambda}[k_P - k_Q(f)]k'_Q(f) = 0,$$

并将式(4.2.4)作为其边界条件。下述方程(4.2.6)中，Q 为轮廓中 C_Q 的参数；$Q'(f)$ 是 C_Q 的正切向量；N_P 是 C_P 的模；k'_Q 是轮廓 C_Q 的曲率的导数。式(4.2.6)的解主要依赖于下列方程中 $f(s)|_{t=0} = f_0(s)$ 初始值的选择：

$$\begin{cases} \dfrac{\partial f(s)}{\partial t} + f''(s)\|Q'[f(s)]\|^2 + k_P < N_P, \\ f(s)|_{t=0} = f_{0(s)}, \end{cases}$$

$$Q'(f) > + \frac{1}{\lambda}[k_P - k_Q(f)]k'_Q(f) = 0。 \tag{4.2.6}$$

为避免 $E(f)$ 局部极小，$f_0(s)$ 应选真实解的较好近似函数。Cohen 采用有限元法解这个方程。

4.2.2 简化方法

显然式(4.2.6)比较复杂，即使用有限元法，求解也困难。为此，可以在保持匹配和弹性约束要求的前提下，设法导出一个简化方程。

式(4.2.3)可重写成：

$$E_{\text{elastic}} = \oint_{C_P} \left\| \lim_{\Delta s \to 0} \left[\frac{\Delta Q(s')}{\Delta s} - \frac{\Delta P(s)}{\Delta s} \right] \right\|^2 \mathrm{d}s,$$

根据三角不等式，有：

$$\left\| \lim_{\Delta s \to 0} \left[\frac{\Delta Q(s')}{\Delta s} - \frac{\Delta P(s)}{\Delta s} \right] \right\|^2 \geqslant \left(\lim_{\Delta s \to 0} \left[\left\| \frac{\Delta Q(s')}{\Delta s} \right\| - \left\| \frac{\Delta P(s)}{\Delta s} \right\| \right] \right)^2, \tag{4.2.7}$$

其中，弧长可近似为

$$\lim_{\Delta s' \to 0} \Delta s' = \| \Delta Q(s') \|,$$
$$\lim_{\Delta s \to 0} \Delta s = \| \Delta P(s) \|。$$

在这个问题中，$s \to s'$ 是连续的一一对应的映射，于是将此关系式代入式(4.2.7)，有

$$\left\| \lim_{\Delta s \to 0} \left[\frac{\Delta Q(s')}{\Delta s} - \frac{\Delta P(s)}{\Delta s} \right] \right\|^2 \geqslant \left| \lim_{\Delta s \to 0} \left[\frac{\Delta s'}{\Delta s} - \frac{\Delta s}{\Delta s} \right] \right|^2 \equiv \left| \frac{\partial [f(s) - s]}{\partial s} \right|^2$$

或

$$\oint_{C_P} \left\| \frac{\partial Q(s')}{\partial s} - \frac{\partial P(s)}{\partial s} \right\|^2 \mathrm{d}s \geqslant \oint_{C_P} \left| \frac{\partial [f(s) - s]}{\partial s} \right|^2 \mathrm{d}s。 \tag{4.2.8}$$

这就是说 $\oint_{C_P} \left[\frac{\partial [f(s) - s]}{\partial s} \right]^2 \mathrm{d}s$ 比式(4.2.3)定义的弹性约束项的约束程度更强。

本文即采用这种简化的弹性约束项：

$$E_{\text{elastic}} = \oint_{C_P} \left[\frac{\partial [f(s) - s]}{\partial s} \right]^2 \mathrm{d}s, \tag{4.2.9}$$

其中：$[f(s) - s]$ 是变形引起的偏移。

使用式(4.2.9)定义的弹性约束项，根据 Euler-Lagrange 方程可得简化方程：

$$\begin{cases} f'' + \dfrac{1}{\lambda} \{ k_P + k_Q[f(s)] \} k'_Q(f) = 0 \\ \text{边界条件式(4.2.4)。} \end{cases} \tag{4.2.10}$$

式(4.2.10)为非线性两点边界值问题，比式(4.2.6)要简单。

现在，讨论两种极限情况：

首先，$\lambda \to \infty$ 表示只考虑弹性约束，方程(4.2.10)成为

$$\begin{cases} f'' = 0 \\ \text{边界条件式(4.2.4)。} \end{cases} \tag{4.2.11}$$

方程的解为 $f = \alpha s$。显然，如果只考虑弹性约束，两轮廓线 C_P 和 C_Q 上匹配点间为线性关系。

另一种极限情况是 $\lambda \to 0$，表明只考虑曲率匹配，这种情况下有：

① 如果 $k'_Q[f(s)] \neq 0$，有 $k_P(s) - k_Q[f(s)] = 0$，即曲率相等时才匹配。

② 如果 $k'_Q[f(s)] = 0$，则方程(4.2.10)等于 0，解没有意义(退化解，C_Q 为一个圆)。

4.3 求解非线性两点边界值问题

对于方程(4.2.10)所描述的非线性两点边界值问题，尤其是用于匹配复杂形状的轮廓线，很难用解析法得到精确解，通常用数值计算法求近似解。对于解方程(4.2.10)，由于常要匹配复杂形状的轮廓线，根据经验采用有限差分法比较容易得到迭代稳定的近似解。为了便于讨

论，方程(4.2.10)可以写成：

$$\frac{\partial^2 f}{\partial s^2} - b \equiv H(s), \tag{4.3.1a}$$

其中：

$$b = -\frac{1}{\lambda}\{k_P(s) - k_Q[f(s)]\}\frac{\partial k_Q[f(s)]}{\partial s}。 \tag{4.3.1b}$$

4.3.1　有限差分法

用有限差分法，将方程(4.2.10)离散化，可得下列方程式：

$$\frac{w_{i-1} - 2w_i + w_{i+1}}{h^2} = b_i \tag{4.3.2}$$

对于 $i=1,2,\cdots,N$，w_i 和 b_i 分别是何在离散点 s_i 处的值。如果轮廓 C_P 的长度为 L_1，轮廓 C_Q 的长度为 L_2，且 $\alpha=L_2/L_1$，则边界条件为：

$$w_0 = 0, \quad w_{N+1} = \alpha。 \tag{4.3.3}$$

这样，可以构成一个 $N\times N$ 非线性方程组：

$$\begin{aligned} -2w_1 + w_2 &= h^2 b_1(s_1,w_1), \\ w_1 - 2w_2 + w_3 &= h^2 b_2(s_2,w_2), \\ &\vdots \\ w_{N-2} - 2w_{N-1} + w_N &= h^2 b_{N-1}(s_{N-1},w_{N-1}), \\ w_{N-1} - 2w_N &= h^2 b_N(s_N,w_N) + \alpha, \end{aligned}$$

写成矩阵形式为

$$\boldsymbol{A}\boldsymbol{w} = h^2 \boldsymbol{b}(\boldsymbol{s},\boldsymbol{w})。 \tag{4.3.4}$$

方程(4.3.1)可重写成

$$H = h^{-2}\boldsymbol{A}\boldsymbol{w} - \boldsymbol{b}(s,\boldsymbol{w}), \tag{4.3.5}$$

其中：

$$\boldsymbol{A} = \begin{vmatrix} -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ & & \vdots & & & \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & 0 & \cdots & 0 & 1 & -2 \end{vmatrix},$$

$$\boldsymbol{w} = [w_1 \quad w_2 \quad \cdots \quad w_N]^{\mathrm{T}},$$

$$b(s,w) = \left[b_1(s_1,w_1) \quad b_2(s_2,w_2) \quad \cdots \quad b_{N-1}(s_{N-1},w_{N-1}) \quad b_N(s_N,w_N) + \frac{\alpha}{h^2}\right]^{\mathrm{T}},$$

式中：T 表示矩阵的转置；$\boldsymbol{A}$ 是一个稀疏三对角矩阵。这个代数方程组的解法在第三章中已作了详细介绍，这里不再赘述。

4.3.2　轮廓线正规化

为了便于处理不同长度的轮廓线 C_Q 和 C_P 的匹配问题，应该在计算曲率前，将轮廓线的长度统一或正规化成一个常数 $\bar{L}$，使匹配算法具有长度不变性。通过使用正规化的轮廓线，式(4.3.4)中的边界条件可简化为：

$$f(1) = 1, \quad (\alpha = 1)。 \tag{4.3.6}$$

4.4　多尺度有限差分法

在曲线匹配计算中，初始的匹配函数估计 f_0 对以后的运算是否收敛有很大影响。对严重变形的轮廓，或有复杂的凹/凸曲线结构的轮廓，线性初始估计 $f(s)=s$ 能导致算法在迭代过程中发散。

采用多尺度的方法能减少算法对初始匹配函数 f_0 的依赖程度。由式(4.2.21)和(4.2.22)所确定的迭代算法，是将 $w^{(k)}$ 通过局部迭代算法得到解。通过对轮廓线 C_P 和 C_Q 进行平滑，曲率 k_Q 和 k_P 的差以及 k_Q 的一阶和两阶导数都会减小。最粗糙(coarsest)时(轮廓线经过高斯滤波器强平滑 $\sigma \gg 1$)，因为 $\partial b/\partial w$ 减小，所以算子 $h^2\mathbf{A}$ 减小，使影响迭代收敛速度的谱半径减小，收敛速度加快。

迭代过程中低频误差项的收敛速度远比高频误差项慢。但是，正是低频部分，由于包括了轮廓线大范围变形所引起的大量数据偏差，是造成迭代过程中收敛速度慢的重要原因。一般来说，要使低频误差项减到足够小，需要经过 N 次迭代(N 为轮廓线离散化中插入的点数)。多尺度有限差分法可看作是解决这个问题的有效方法。通过把上一步在较粗糙程度得到的结果作为下一步较精确轮廓线迭代的预估值，能提高轮廓线匹配的速度并避免匹配计算过程中的迭代发散。用这种方法仅需在较粗糙的轮廓线上去估算 f_0，因而，编程比较容易，运算速度较快。

4.5　实验验证

4.5.1　简单的轮廓线匹配

图4.1中轮廓线 C_Q 是正方形经过平移，旋转和长度变形后生成的一个矩形轮廓线 C_P。轮廓线 C_Q 的曲率 k_Q 与轮廓线 C_Q 的曲率 k_P 相比较，在 A 和 B 点有一个较小的偏移。

图4.2表示 $\boldsymbol{b}(\boldsymbol{s},\boldsymbol{w}^{(k-1)})$ 项的意义：图4.2(a)是曲率 k_Q 与 k_P 的差，但这差值还不足以确定匹配过程中轮廓线 C_Q 的局部扩展方向。在公式中轮廓线 C_Q 的扩展方向是由曲率 k_Q 与 k_P

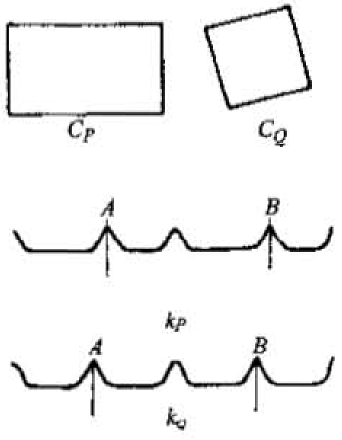


图4.1　两条轮廓线及其曲率

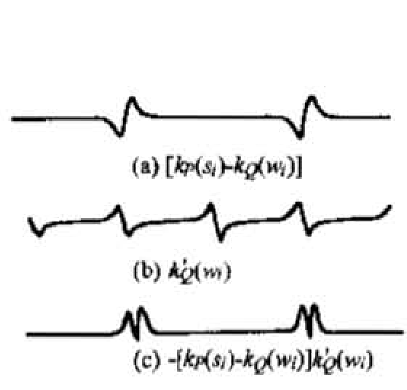


图4.2　$\boldsymbol{b}(\boldsymbol{s},\boldsymbol{w}^{(k-1)})$ 项的意义

的差与 k_P 的一阶导数 k'_P[见图 4.2(b)]的积确定的。图 4.2(c)表示在整条轮廓线上有两个区域中 $b(s, w^{(k-1)})$的值为正。这意味着曲线在这两个区域中应朝坐标轴正方向扩展。通常，如果某个区域中 $b(s, w^{(k-1)})$的值为正(或负)，则表明该区域的轮廓线在匹配过程中局部向正(或负)方向扩展。

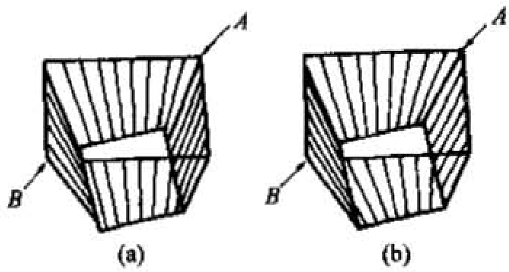


图 4.3 匹配前(a)和匹配后(b)两条轮廓线的对应关系

在初始阶段，假设轮廓线 C_P 和 C_Q 的匹配关系 $f(s)$是线性的。从图 4.3(a)中可以看到，轮廓线 C_P 上的点 A 和 B 与轮廓线 C_Q 上相应的点 A 和 B 不匹配。经过轮廓线匹配法计算，在顶角 A 和 B 都能得到很好的匹配[见图 4.3(b)]。而图 4.4 是经过轮廓线匹配运算后，匹配关系 $f(s)$的最后结果。

4.5.2 多尺度算法

图 4.5～4.7 显示了 Ueda[3] 论文中轮廓线的匹配结果。图的左边给出了曲率 k_Q 与 k_P。

一般多尺度轮廓线匹配方法中，两条轮廓线的结构不一样会导致失配。对基于轮廓线特征表示的链码算法，前一步的失配会给以后匹配运算造成致命影响，因为它递归地把轮廓的匹配部分链码分解成更小的子链码。而基于曲率表示的轮廓线匹配方法，如果轮廓上有一些小的凹/凸不一致，匹配算法仍然有效。这是因为式(4.2.2)指标函数 $E(f)$中平滑约束项的作用。

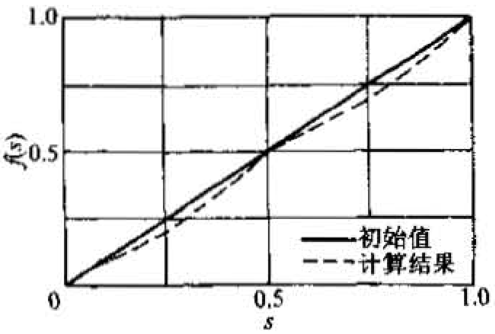


图 4.4 匹配前后 $f(s)$关系

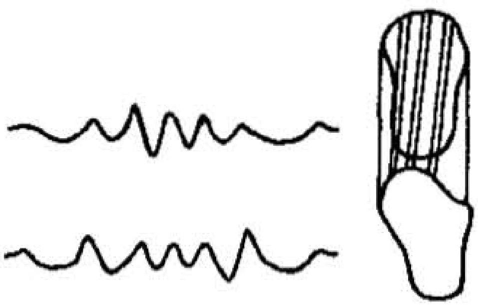


图 4.5 在粗尺度 $\sigma_0=8$，第一步匹配

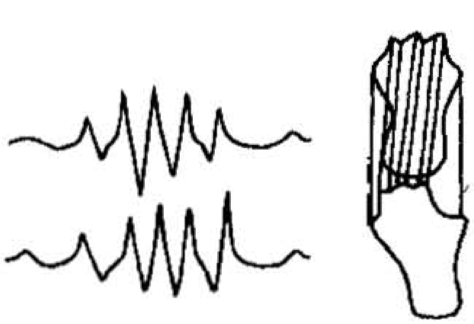


图 4.6 在尺度 $\sigma_2=5$，第二步匹配
(上次匹配的结果作为本次匹配的初值 f_0)

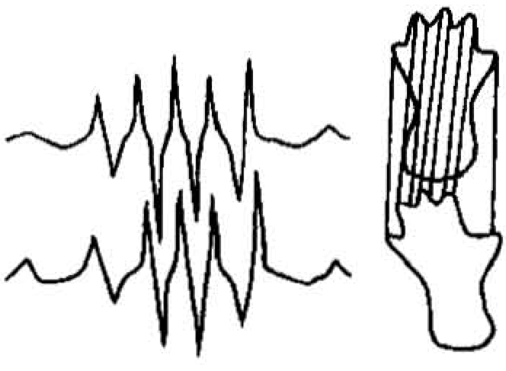


图 4.7 在精细尺度 $\sigma_f=2$，第三步匹配
(上次匹配的结果作为本次匹配的初值 f_0)

4.5.3　大脑解剖图上轮廓线和核磁共振图分割后的轮廓线匹配

大脑解剖图上轮廓线和日常门诊中所得的核磁共振图上分割后的轮廓线的匹配结果见图 4.8 和图 4.9，在经过匹配计算的 13 个不同的脑组织轮廓线中，列出了有代表性的一个结果。

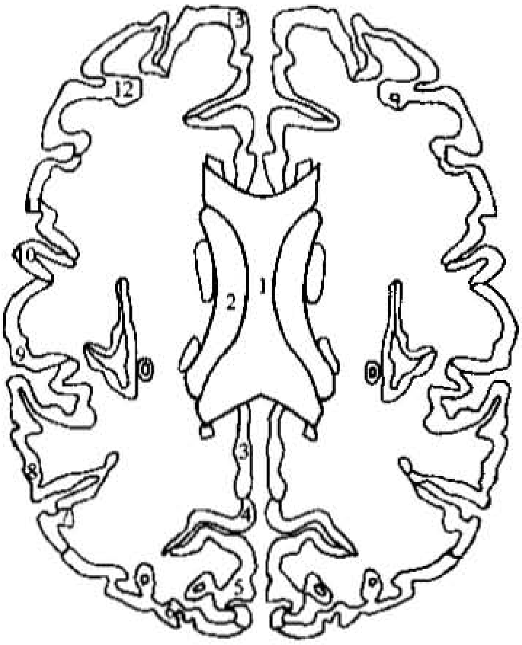


图 4.8　大脑解剖图上轮廓线

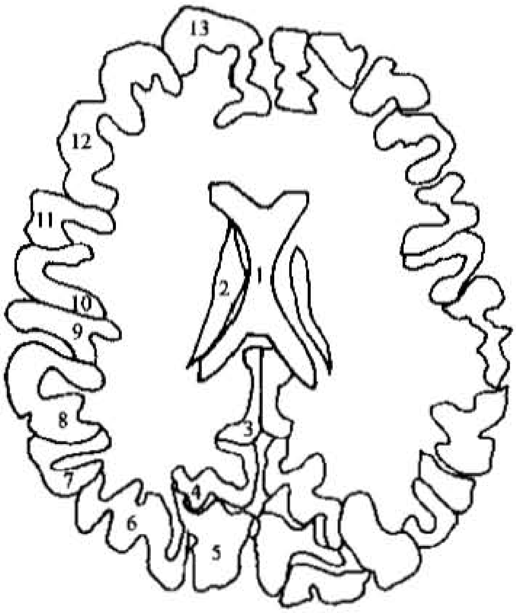


图 4.9　核磁共振图上分割后的轮廓线

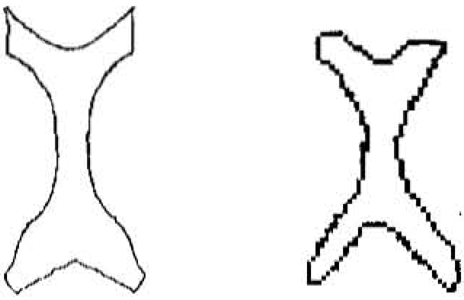


图 4.10　一组对应的脑组织

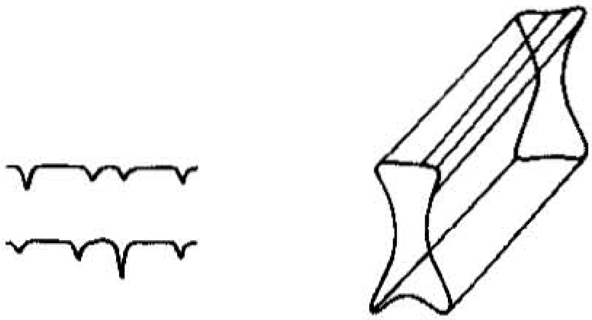


图 4.11　在尺度 $\sigma_l=6$ 匹配结果

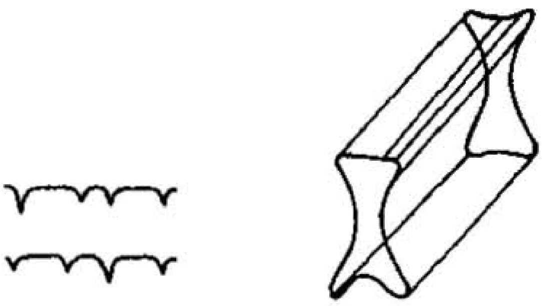


图 4.12　在尺度 $\sigma_l=4$ 匹配结果

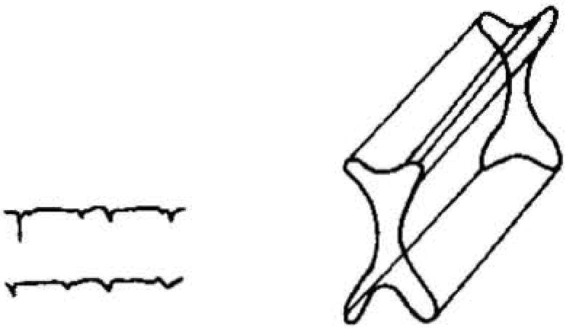


图 4.13　在精细尺度 $\sigma_l=2$ 匹配结果

对于图 4.10 中的脑组织，采用了按顺序逐步减小的多尺度平滑法，即从 $\sigma=6$ 开始每次减少 2，直至 $\sigma=2$。实验证明，即使两条轮廓线之间变形比较大，用多尺度平滑匹配法仍能在某一尺度得到匹配，这个尺度给出了两条轮廓线之间相似程度的信息。见图 4.11～图 4.13。

参 考 文 献

[1] Duncan J S, Lee F A, Smeulders A W M et al. A Bending Energy Model for Measurement of Cardiac Shape Deformity. IEEE trans. on Medical Imaging, vol. 10, no. 3, pp. 307-320, Sept., 1991

[2] Mokhtarian F and Mackworth A. Scale-based description and recognition of planar curves and two-dimensional shapes. IEEE Trans. Pattern Anal. Machine Intell. vol. PAMI-8, no. 1, pp. 34-43, Jan., 1986

[3] Ueda N, Satoshi S. Learning Visual Models from shape Contours Using Multiscale Convex/Concave Structure Matching. IEEE Trans. Pattern Anal. Machine Intell. vol. PAMI-15, no. 4, pp. 337-352, April, 1993

[4] Wolfson H J. On Curve Matching. IEEE Trans. Pattern Anal. Machine Inell., vol. PAMI-12, no. 5, pp. 483-489, May, 1990

[5] Xin Y, Truyen B, Pratikakis I, et al. Hierarchical Contour Matching in Medical Images. Image and Vision Computing, vol. 14, no. 6, pp. 417-433, 1996

[6] Yuille A L, Poggio T A. Scaling Theorems for Zero Crossings. IEEE trans. Pattern Anal. Machine Intell. vol. PAMI-8, no. 1, pp. 15-25, Jan., 1986

第5章　运动估计

计算机视觉和图像处理中，关于多帧或序列图像处理技术，因其在医学图像处理和传输、机器人视觉以及航空航天制导等领域都有着广泛的应用，已成为许多学者关注的热点。而序列图像处理的核心技术之一是运动估计和图像匹配技术，运动估计和图像匹配的数学模型相似，可以相互借鉴。

所谓序列图像的运动估计，是根据时间序列图像中的两帧或多帧图像，估计出图像中各目标的运动信息。序列图像运动估计及在此基础上的运动补偿技术，是利用序列图像帧间的相关性，消除冗余度，广泛地应用于图像压缩编码、图像传输等领域。图像配准，就是将多幅相互关联的图进行比较，以确定图像间目标或点的对应关系或相对变形。这些研究可应用于运动目标跟踪、图像中特定目标的检测和精确定位，例如：医学上心脏图像的动态分析，军事上运动目标跟踪以及遥感图的分析等。

本章针对现有运动场估计技术在处理较复杂的形变图像序列(特别是有弹性形变的图像)时存在的问题，提出一种基于位移矢量场的运动估计方法。这方法也是上一章轮廓线匹配工作在二维图像空间的扩展。

现有的运动场估计技术，按所使用的图像信息不同分为两大类：基于特征的方法和直接利用图像灰度信息等原始数据的方法。

基于图像中目标特征的方法，也称离散处理的方法，是指匹配前先提取图像中目标的某些特征或链码描述，然后进行图像匹配的一类算法。如果目标的特征或链码描述容易提取，那么这种方法的计算速度是很快的。基于特征的运动场估计方法能找出图像中位移量比较大的目标。然而，这一方法对于不同的目标常需要选择不同的特征，实时匹配时间序列图像时就不太容易。在多目标环境下，这种方法需解决目标的对应问题。此法有两大缺点：

(1) 易受图像畸变等原因的影响。

(2) 序列图像间根据特征建立目标对应关系比较困难。

另外基于特征的方法如何选取有效的特征，以及如何根据得到各特征点的稀疏的运动场信息，真止建立起整个图像的运动场都是这一方法需要解决的问题。

直接利用图像灰度信息的方法的基本数学模型为：

$$g(x,y,t) = a \cdot g(x+\delta x, y+\delta y, t+\delta t) + m + n,$$

式中：$g(x,y,t)$表示序列图像t时刻在(x,y)位置的灰度；δx和δy分别是(x,y)位置的目标点在时间间隔δt后x和y位置发生的变化；参数a是考虑到图像光照度不同而引入的尺度因子；m是修正灰度值偏差的偏移项；n是噪声项；a和m可看作是因光照和图像传输原因导致的畸变，与$(\delta x,\delta y)$及时间项有关。因光照和图像传输原因导致的畸变问题不是本书叙述的重点，所以在位移场估计问题中，只考虑简单模型，即不考虑光照及传输等使图像发生畸变因数，认为序列图像的各帧中物体在运动轨迹上灰度不变。

直接利用图像灰度信息的方法又可分为三大类：块匹配法、光流法、像素递归法。

这里介绍的位移场估计算法将一维轮廓线弹性匹配的思想运用到二维图像空间，日的是希望能检测复杂的弹性形变。

本算法的主要特点是：

(1) 算法得到的运动场是位移场，而不是通过光流法计算得到的某种近似速度场。因此本算法不需做时间标定，可直接得到图像间各点空间对应关系，便于做运动补偿和恢复。另外作为匹配性能定量指标之一的匹配残差的计算更为容易和合理。

(2) 与以块匹配法相比，此方法对有缩放、扭曲等复杂形变的匹配更为有效。

(3) 但是本算法与大多基于图像灰度的梯度算法一样，计算时间比较长，而且序列图像中目标的位移量不允许过大。

5.1 数学模型

为了进行位移场的估计，需要在数学模型中构造能量函数。这个能量函数由两项组成：匹配误差项和平滑约束条件。通过最小化这个能量函数得到序列图像的位移场。这能量函数基于两个类似于光流法中应用的假设：沿运动方向图像灰度保持不变和图像位移场基本平滑变化。

考虑图像序列中的各帧 $G\subset\Re^2$，$\Re^2$ 是二维欧几里德空间。设 G_2 是 G_1 经过弹性变形而得到的。设 G_1 是某一图像序列中的第一帧图像，$g_1(\boldsymbol{r})$是表示第一帧图像 G_1 在空间位置 $\boldsymbol{r}=(x,y)^t$ 处的灰度值，其中 t 表示矢量的转置。对于黑白图像灰度值的范围为 $g=\{i\mid 0\leqslant i\leqslant 255\}$，不失一般性，取图像 G_1 为

$$G_1=\{(x,y)^t\mid 0\leqslant x\leqslant 1,0\leqslant y\leqslant 1\};$$

同样，图像 G_2 为

$$G_2=\{(X,Y)^t\mid 0\leqslant X\leqslant a,0\leqslant Y\leqslant b\}。$$

这里，变形限定于弹性变形，即假定第二帧图像 G_2 经一些变形后可恢复成其参考图像 G_1。这就是说，这里不考虑重叠、阴影和光照不均匀等因素。

事实上，要解决的匹配问题就是要判断图像 G_1 中的每个点 $\boldsymbol{r}$ 在 G_2 中所相应的点 $\boldsymbol{r}'$。设 f 是定义于 $L^2(\Re^2)$上的连续可微的函数，$L^2(\Re^2)$表示 $\Re^2$ 上平方可积可测函数组成的矢量空间。其中，$f:\boldsymbol{r}'=f(\boldsymbol{r})$或 $f(\boldsymbol{r})=[X(x,y),Y(x,y)]$。

基于这一目的，定义包含 $X=X(x,y)$和 $Y=Y(x,y)$的能量函数：

$$E[X,Y]=E_{\text{match}}[X,Y]+\alpha E_{\text{sm}}[X,Y],\tag{5.1.1}$$

其中：E_{match}为第一帧图像与位移后的第二帧图像的相似性测度；$E_{\text{sm}}[X,Y]$为位移场平滑性测度；参数 α 为平滑项在能量方程中的权重，α 越大表示增强对位移场的平滑。

E_{match}定义为 $g_1(\boldsymbol{r})$与 $g_2(f(\boldsymbol{r}))$间差值的平方：

$$E_{\text{match}}[X,Y]=\iint_G\{g_1(x,y)-g_2[X(x,y),Y(x,y)]\}^2\mathrm{d}x\mathrm{d}y。\tag{5.1.2}$$

平滑项 E_{sm}的引入是为解决“瓶颈问题”，即一个方程解两个未知量的问题。类似于光流法中的平滑约束条件，本位移场估计算法的平滑项为

$$E_{\text{sm}}=\iint_G\left\{\left[\frac{\partial(X-x)}{\partial x}\right]^2+\left[\frac{\partial(X-x)}{\partial y}\right]^2+\left[\frac{\partial(Y-y)}{\partial x}\right]^2+\left[\frac{\partial(Y-y)}{\partial y}\right]^2\right\}\mathrm{d}x\mathrm{d}y。\tag{5.1.3}$$

最小化式(5.1.1)中的能量函数 $E(X,Y)$，即可得到两帧图像的对应关系函数 $f(\boldsymbol{r})$。

5.2 匹配问题的变分分析

对于式(5.1.1)中所建立的能量函数和求其极值问题，在数学上称为泛函的极值问题。在

1.8节中已经作了介绍。这里具体讨论如何用变分原理解这个问题。二维的 Euler 方程表示为

$$\frac{\partial E}{\partial X}-\frac{\partial}{\partial x}\left(\frac{\partial E}{\partial x}\right)-\frac{\partial}{\partial y}\left(\frac{\partial E}{\partial y}\right)=0 \tag{5.2.1}$$

和

$$\frac{\partial E}{\partial Y}-\frac{\partial}{\partial x}\left(\frac{\partial E}{\partial x}\right)-\frac{\partial}{\partial y}\left(\frac{\partial E}{\partial y}\right)=0。\tag{5.2.2}$$

为最小化能量函数(5.1.1),将式(5.1.2)(5.1.3)代入式(5.1.1),得到

$$\begin{aligned}E[X,Y]=\iint_G\Big[&\{g_1(x,y)-g_2[X(x,y),Y(x,y)]\}^2\\&+\alpha\left\{\left[\frac{\partial(X-x)}{\partial x}\right]^2+\left[\frac{\partial(X-x)}{\partial y}\right]^2+\left[\frac{\partial(Y-y)}{\partial x}\right]^2+\left[\frac{\partial(Y-y)}{\partial y}\right]^2\right\}\Big]\mathrm{d}x\mathrm{d}y。\end{aligned}\tag{5.2.3}$$

由于 $X(x,y),Y(x,y)$ 相互正交,为最小化能量函数(5.2.1),对 X 和 Y 分别用 Euler 方程(5.2.1)和(5.2.2)。由式(5.2.1)得到

$$2[g_2(X,Y)-g_1(x,y)]\frac{\partial g_2}{\partial X}-\frac{\partial}{\partial x}\left(2\alpha\frac{\partial X}{\partial x}\right)-\frac{\partial}{\partial y}\left(2\alpha\frac{\partial X}{\partial y}\right)=0。\tag{5.2.4}$$

重新整理式(5.2.4)得到

$$[g_2(X,Y)-g_1(x,y)]\frac{\partial g_2}{\partial X}-\alpha\left(\frac{\partial^2 X}{\partial x^2}+\frac{\partial^2 X}{\partial y^2}\right)=0。\tag{5.2.5}$$

同样,对 Y 用 Euler 方程(5.2.2),得

$$\begin{cases}[g_2(X,Y)-g_1(x,y)]\dfrac{\partial g_2}{\partial X}-\alpha\nabla^2 X=0,\\[g_2(X,Y)-g_1(x,y)]\dfrac{\partial g_2}{\partial Y}-\alpha\nabla^2 Y=0,\\\text{边界条件。}\end{cases}\tag{5.2.6}$$

其中:∇^2X,∇^2Y 是 X、Y 的拉普拉斯方程 $\nabla^2X=\frac{\partial^2X}{\partial x^2}+\frac{\partial^2X}{\partial y^2}$,$\nabla^2Y=\frac{\partial^2Y}{\partial x^2}+\frac{\partial^2Y}{\partial y^2}$。为求唯一解,还需加边界条件。因而最小化能量函数(5.1.1),X、Y 需满足方程(5.2.6)。式(5.2.6)是一个典型的椭圆方程,这样匹配或位移估计问题的变分解属于椭圆方程的边值问题。

5.3 边界条件

关于图像位移场估计及匹配问题的边界条件,常用两种边界条件:一种称之为弯曲的边界条件(Bending boundary condition);另一种称之为滑动的边界条件(Sliding boundary condition)。图 5.1 给出满足这两种边界条件有形变的二维图形。

如设二维模板中的点 $\boldsymbol{r}=(x,y)$ 对应形变后模板位置为 $\boldsymbol{r}'=[X(x,y),Y(x,y)]$,设 $\boldsymbol{S}(x,y)$ 是这两个对应点间的位移,$\boldsymbol{S}(x,y)=(S_x,S_y)=\boldsymbol{r}'-\boldsymbol{r}$,则弯曲的边界条件可用下列数学公式表达:

$$\begin{cases}\left.\dfrac{\partial S_x}{\partial x}\right|_{(0,y)}=\left.\dfrac{\partial S_x}{\partial x}\right|_{(1,y)}=\left.\dfrac{\partial S_y}{\partial y}\right|_{(x,0)}=\left.\dfrac{\partial S_y}{\partial y}\right|_{(x,1)}=0,\\S_x(x,0)=S_x(x,1)=S_y(0,y)=S_y(1,y)=0。\end{cases}\tag{5.3.1}$$

由公式知边界条件由 Neumann 边界条件和 Dirichlet 边界条件组成。这种边界条件使形变模

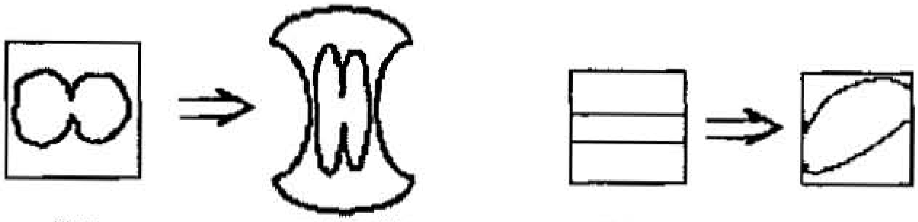


图 5.1 两种边界条件约束的图像形变示例

(a) 弯曲的边界条件约束的图像形变；(b) 滑动的边界条件约束的图像形变

板图像的 4 个角点位置锁定在原模板对应位置，不发生变化，允许边界有较灵活的形变。

滑动的边界条件同样由 Neumann 边界条件和 Dirichlet 边界条件组成，可用下列数学公式表达：

$$\begin{cases}\left.\dfrac{\partial S_x}{\partial y}\right|_{(x,0)}=\left.\dfrac{\partial S_x}{\partial y}\right|_{(x,1)}=\left.\dfrac{\partial S_y}{\partial x}\right|_{(0,y)}=\left.\dfrac{\partial S_y}{\partial y}\right|_{(1,y)}=0,\\ S_x(0,y)=S_x(1,y)=S_y(x,0)=S_y(x,1)=0。\end{cases}\tag{5.3.2}$$

这种边界条件，使形变后模板的边界对应于参考模板的边界，因为用这种边界条件模板边界上的点可沿边界滑动，所以称之为滑动边界条件。

本文中假设两帧图像在边界上的点不发生任何变形（包括滑动）或仅有线性变形（当图像大小发生变化时）。边界条件可直接通过对应点位置表示为

$$\begin{cases}X(0,y)=0, & X(1,y)=a, & \forall y\in[0,1];\\ X(x,0)=ax, & X(x,1)=ax, & \forall x\in[0,1];\\ Y(x,0)=0, & Y(x,1)=b, & \forall x\in[0,1];\\ Y(0,y)=by, & Y(1,y)=by, & \forall y\in[0,1]。\end{cases}\tag{5.3.3}$$

这里图像第一帧和第二帧图像的大小分别是 1×1 和 $a\times b$。如 $a=b=1$，则表示匹配图像边界上的点不发生任何变形（见图 5.2）。

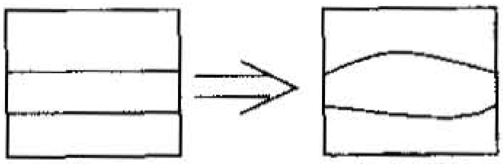


图 5.2 受本文边界条件约束的图像

5.4 有限差分解法

因为本章位移场估计算法的数学模型经变分分析后得到的是一个典型的椭圆型偏微分方程的边值问题，所以本节首先介绍用于椭圆型偏微分方程的边值问题的几种典型利用有限差分的数值解法，并使用同步超松弛法（S. O. R.）解本文位移场估计问题。

5.4.1 用差分代替导数的方法

用差分方法解微分方程的边值问题的第一步是将求解区域进行剖分（如图 5.3 所示）。为简单起见，沿与方向的网格的步长都取为 h。图中“·”为内网格点。

对于光滑的函数 X 和 Y，由 Taylor 展开式得

$$\left[\frac{\mathrm{d}X}{\mathrm{d}x}\right]_{i,j}=\frac{U_{i+1,j}-U_{i-1,j}}{2h}+O(h^2),\tag{5.4.1}$$

$$\left[\frac{\mathrm{d}Y}{\mathrm{d}x}\right]_{i,j} = \frac{V_{i+1,j} - V_{i-1,j}}{2h} + O(h^2)。 \qquad (5.4.2)$$

这里$(U_{i,j}, V_{i,j})$分别表示函数 $X(x,y)$和 $Y(x,y)$在$(x=i, y=j)$处的离散近似值,$O(h^2)$是 h^2 的同阶无穷小。

将图 G_2 分割成网格图,例如,G_2 被划分成$(M+1)\times(N+1)$的网格图,节点坐标分别是 $x_i=ih, y_j=jh$,这里,$i=0,1,2,\cdots,M+1$;$j=0,1,2,\cdots,N+1$。

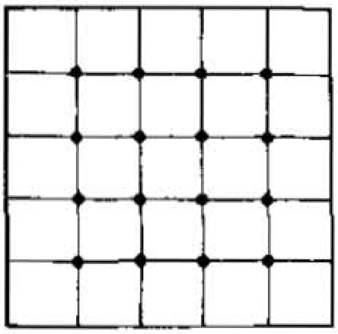


图 5.3　图像网格剖分

当式(5.2.6)的解 X、Y 充分平滑,则沿 x 和 y 分别用中心差商(5.4.1)和(5.4.2)代替导数,且忽略高阶无穷小,则式(5.2.6)中 X、Y 的 Laplace 算子的离散形式为

$$\nabla^2 X = \frac{1}{h^2}(U_{i,j+1} + U_{i+1,j} + U_{i-1,j} + U_{i,j-1} - 4U_{i,j}), \qquad (5.4.3)$$

$$\nabla^2 Y = \frac{1}{h^2}(V_{i,j+1} + V_{i-1,j} + V_{i+1,j} + V_{i,j-1} - 4V_{i,j})。 \qquad (5.4.4)$$

上式因差分方程中只出现网格 U、V 在其上、下、左、右四个邻点上的值,故称之为五点差分格式。又可得

$$B_{i,j} = [g_2(X_{i,j}, Y_{i,j}) - g_1(i,j)]\left.\frac{\partial g_2}{\partial X}\right|_{\substack{X=X_{i,j}\\Y=Y_{i,j}}} \qquad (5.4.5)$$

$$= [g_2(X_{i,j}, Y_{i,j}) - g_1(i,j)][g_2(X_{i+1,j}, Y_{i+1,j}) - g_2(X_{i-1,j}, Y_{i-1,j})]/2h,$$

$$C_{i,j} = [g_2(X_{i,j}, Y_{i,j}) - g_1(i,j)]\left.\frac{\partial g_2}{\partial Y}\right|_{\substack{X=X_{i,j}\\Y=Y_{i,j}}} \qquad (5.4.6)$$

$$= [g_2(X_{i,j}, Y_{i,j}) - g_1(i,j)][g_2(Y_{i,j+1}, Y_{i,j-1}) - g_2(X_{i,j-1}, Y_{i,j-1})]/2h,$$

将式(5.4.5)、(5.4.6)代入式(5.2.6),得到最小化能量方程应满足方程的离散形式:

$$\begin{cases} U_{i,j+1} + U_{i-1,j} + U_{i+1,j} + U_{i,j-1} - 4U_{i,j} = \dfrac{hB_{i,j}}{\alpha}, \\ V_{i,j+1} + V_{i-1,j} + V_{i-1,j} + V_{i,j-1} - 4V_{i,j} = \dfrac{hC_{i,j}}{\alpha}。 \end{cases} \qquad (5.4.7)$$

边界条件式的离散形式为:

$$\begin{cases} U_{0,j}=0; \quad U_{M-1,j}=a, \quad j=1,2,\cdots,N-1; \\ U_{i,0}=\dfrac{ai}{M}; \quad U_{i,N-1}=\dfrac{ai}{M}, \quad i=1,2,\cdots,M-1; \\ V_{i,0}=0; \quad V_{i,N-1}=\dfrac{ai}{M}, \quad i=1,2,\cdots,M-1; \\ V_{0,j}=\dfrac{bj}{N}; \quad V_{M-1,j}=\dfrac{bj}{N}, \quad j=1,2,\cdots,N-1。 \end{cases} \qquad (5.4.8)$$

式(5.4.7)和(5.4.8)可写成矩阵形式:

$$\begin{cases} \boldsymbol{AU} = \dfrac{h}{\alpha}\boldsymbol{B} - \boldsymbol{G}, \\ \boldsymbol{AV} = \dfrac{h}{\alpha}\boldsymbol{C} - \boldsymbol{P}。 \end{cases} \qquad (5.4.9)$$

其中：$\boldsymbol{U}=\begin{bmatrix} U_{1,1} \\ \vdots \\ U_{1,N} \\ U_{2,1} \\ \vdots \\ U_{M,N} \end{bmatrix}$，$\boldsymbol{V}=\begin{bmatrix} V_{1,1} \\ \vdots \\ V_{1,N} \\ V_{2,1} \\ \vdots \\ V_{M,N} \end{bmatrix}$，$\boldsymbol{G}$ 和 $\boldsymbol{P}$ 由边界条件(5.4.8)得到，

$$\boldsymbol{A}=\begin{bmatrix}
-4 & 1 & & \cdots & & 1 & & & & \\
1 & -4 & 1 & & & & 1 & & & \\
 & & \ddots & & & & & \ddots & & \\
\vdots & & 1 & -4 & 1 & & & & 1 & \\
 & & & 1 & -4 & & & & & 1 \\
1 & & & & & -4 & 1 & & & \\
 & 1 & & & & 1 & -4 & 1 & & \\
 & & \ddots & & & & & \ddots & & \\
 & & & 1 & & & & 1 & -4 & 1 \\
 & & & & 1 & & \cdots & & 1 & -4
\end{bmatrix}。$$

5.4.2 松弛算法解差分方程

对于本章问题，直接解 U 和 V 比较困难，但通过迭代可较好地解出有限差分方程(5.4.7)的代数解。在第 2 章介绍的各种迭代方法中，由于实现比较容易，松弛算法(S.O.R.)得到广泛的应用。

(1) 设 U^0 和 V^0 是 U 和 V 的初始值，则

$$\begin{cases} U^0_{i,j} = \dfrac{ai}{M}, \\ V^0_{i,j} = \dfrac{bj}{N}, \end{cases} \quad i = 1,2,\cdots,M-2, j = 1,2,\cdots,N-2。 \tag{5.4.10}$$

式(5.4.10)是线性的，即没有变形时的初始值。

(2) 将 $U^{k-1}_{i,j}$ 和 $V^{k-1}_{i,j}$ 代入式(5.4.7)，可得到 $B_{i,j}$ 和 $C_{i,j}$。这里 $U^{k-1}_{i,j}$ 和 $V^{k-1}_{i,j}$ 是上一次(第 $k-1$ 次)迭代的结果，开始时，$k-1$ 等于零。

(3) 通过 Gauss-Seidel 法解方程(5.4.7)。设 $\widetilde{U}^{k}_{i,j}$ 和 $\widetilde{V}^{k}_{i,j}$ 是第 k 次 Gauss-Seidel 迭代的结果，即：

$$\begin{cases} \widetilde{U}^{k}_{i,j} = -\dfrac{h}{4\alpha}B_{i,j} + \dfrac{1}{4}U^{k-1}_{i+1,j} + \dfrac{1}{4}U^{k-1}_{i,j+1} + \dfrac{1}{4}\widetilde{U}^{k}_{i-1,j} + \dfrac{1}{4}\widetilde{U}^{k}_{i,j-1}, \\ \widetilde{V}^{k}_{i,j} = -\dfrac{h}{4\alpha}C_{i,j} + \dfrac{1}{4}V^{k-1}_{i+1,j} + \dfrac{1}{4}V^{k-1}_{i,j+1} + \dfrac{1}{4}\widetilde{V}^{k}_{i-1,j} + \dfrac{1}{4}\widetilde{V}^{k}_{i,j-1}。 \end{cases} \tag{5.4.11}$$

(4) Gauss-Seidel 迭代法的收敛速度较慢，因此，通常求 $\widetilde{U}^{k}_{i,j}$ 和 $\widetilde{V}^{k}_{i,j}$ 时使用超松弛法。通过同 Gauss-Seidel 法得到的前一次迭代结果作线性外延，对初始估计逐次修正(为减小误差)，得到一个理想的解，其所需的迭代次数多数情况下可以减少。这一方法可表示为

$$\begin{cases} U^{k}_{i,j} = (1-\omega)U^{k-1}_{i,j} + \omega\,\widetilde{U}^{k}_{i,j}, \\ V^{k}_{i,j} = (1-\omega)V^{k-1}_{i,j} + \omega\,\widetilde{V}^{k}_{i,j}, \end{cases} \quad i = 1,2,\cdots,M-1; j = 1,2,\cdots,N-1, \tag{5.4.12}$$

其中：$0<\omega<2$。第 k 次的值是由 Gauss-Seidel 法得到的值和前一次值外推而得。如果 $\omega=1$，则这一方法蜕变为 Gauss-Seidel 法。参数 ω 与迭代速度有关，它决定了收敛速度。

5.4.3 迭代停止条件

迭代的停止条件可取迭代次数大于或等于某设定值，或同时满足下式：

$$\sum_{x=0}^{N-1}\sum_{y=0}^{M-1}\frac{|X_{n+1}(x,y)-X_n(x,y)|}{MN}<\varepsilon, \tag{5.4.13}$$

$$\sum_{x=0}^{N-1}\sum_{y=0}^{M-1}\frac{|Y_{n-1}(x,y)-Y_n(x,y)|}{MN}<\varepsilon, \tag{5.4.14}$$

其中：ε 为一个小的正数。为了防止迭代时间过长，程序中还是需要检测是否超过最大迭代次数。

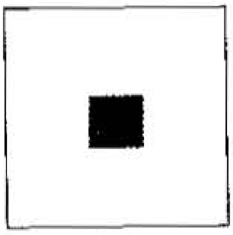
5.5 参数选择

本节讨论匹配算法中的两个重要参数的选择方法：平滑系数 α 和松弛迭代中的松弛系数 ω 的选择。在 α 的选择方法中讨论了 α 对位移场的影响，并进一步给出 Nagel 关于 α 是迭代次数函数的策略，应用并讨论了这一策略的意义。对 ω 的选择讨论了 ω 对收敛速度和迭代稳定性的影响。

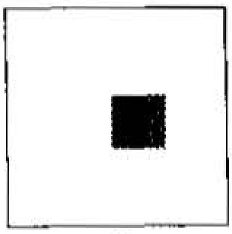
5.5.1 α 的选择

由 5.1 节讨论可知，α 是平滑项在整个能量方程中的权重，α 越大表示图像中位移场越平滑。可根据图像运动场的一致性估计选择 α 的大小，建议 α 的范围在[10,300]区间，本文所用 α 在这个区间。

为比较选择 α 对位移场的影响，用一块正方形位移的图 5.4 作为测试图像，图中的正方形仅向右平移，这样只比较 X 方向的位移。



(a)



(b)

图 5.4 人工合成的有向右平移的块测试图像

(a) 第一帧；(b) 第二帧

图 5.5 给出测试图 5.4 在选用不同的 α 时，得到的位移场中的 X 方向分量的表示。三幅图所用的参数都为 $\omega=0.5$。由图容易看出随着平滑系数 α 的增大，位移场越来越平滑。

关于 α 的选择，Nagel 给出一种分段变化的 α 的选择策略。这种启发式改善 α 选择的方法是：α 是迭代次数的一个函数，函数通过一个手工曲线描述，特点是：起始时，α 选择一个较小但不为 0 的值；而后迭代几次后逐渐变大，至一定次数后为最终恒定值。

图 5.7 是将 α 取常数($\alpha=50$)与分段变化(见图 5.6)得到的残差收敛曲线的比较。测试图用图 5.4 的匹配结果，其中 $\omega=0.5$。由图 5.7 可知，采用变化的 α 策略，匹配的收敛有所加

快，收敛后平均匹配残差结果与 α 为常数时基本相同。但因为此 α 变化的曲线与图像本身关系密切，需不断调整，所以本文后面章节中，仍采用 α 为常数的方法。

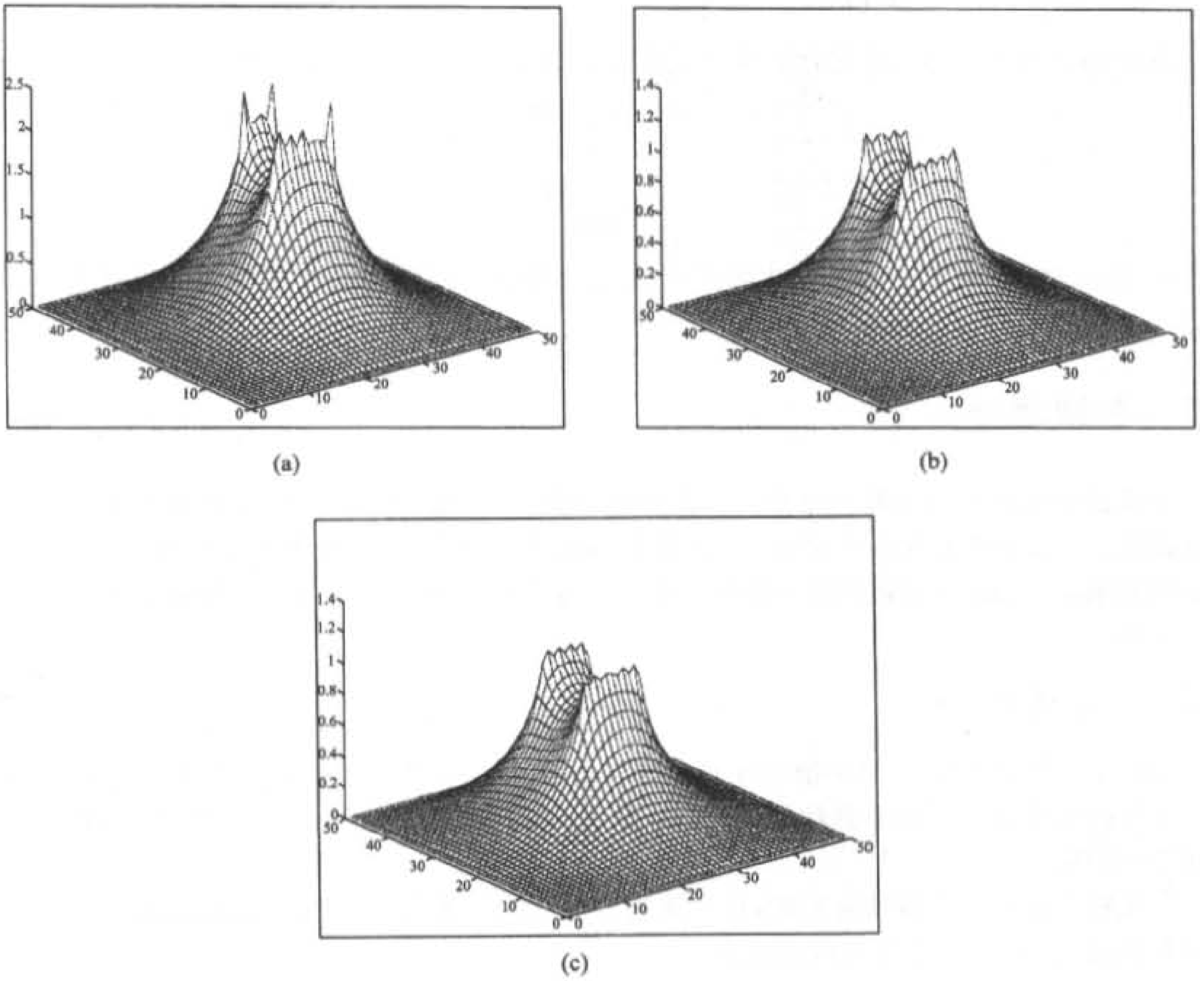


图 5.5 测试图 5.4 在选用不同的 α 时，得到的位移场中的 X 方向分量在 Matlab 中的表示

(a) $\alpha=50$ 时得到的位移场 X 方向分量；(b) $\alpha=100$ 时得到的位移场 X 方向分量；

(c) $\alpha=200$ 时得到的位移场 X 方向分量

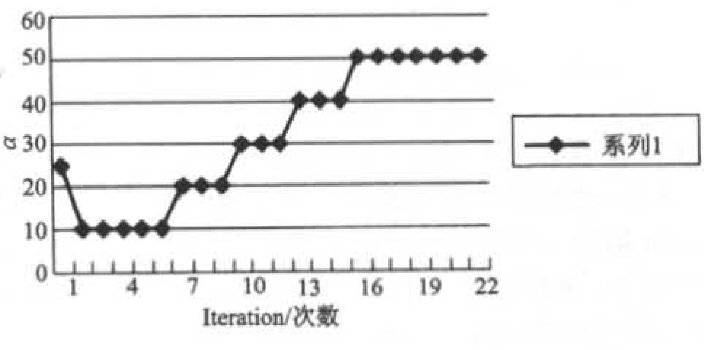


图 5.6 分段变化的 α 的变化曲线

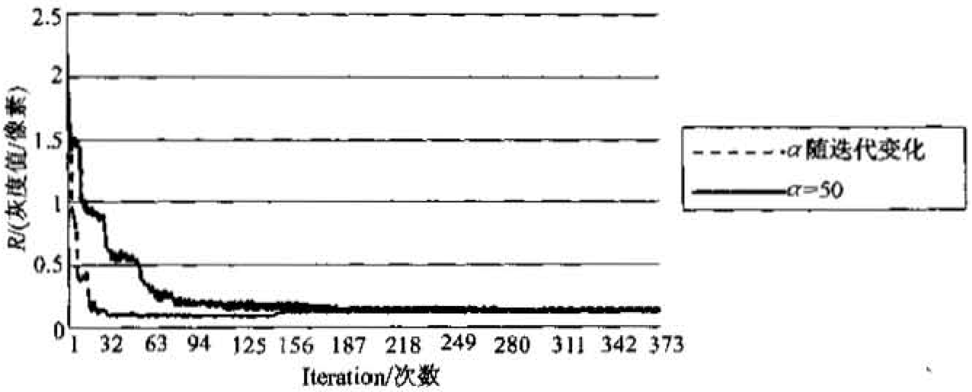


图 5.7　匹配测试图 5.4 当 $\alpha=50$ 时和分段变化的 α 的收敛曲线比较($\omega=0.5$)

尽管如此，采用 α 随迭代次数变化的策略，给我们有益的启示，即在匹配图像的迭代计算中，迭代开始阶段，α 不应取得太大，否则将减缓计算的速度，应使平滑系数随迭代次数逐渐增加。

5.5.2　ω 的选择

为了算法可行，迭代算法不但应该收敛，而且收敛速度必须足够快。因此需要确定较优的 ω_{opt}，当 ω 接近于 ω_{opt} 时，式(5.4.12)收敛速度达到最快。一般来说，确定 ω_{opt} 是困难的，通常是通过不断的试验和判断误差减小的方法来获得 ω_{opt}。可以证明，当 $\omega\leqslant\omega_{opt}$ 时收敛是单调的，而 $\omega>\omega_{opt}$ 时是振荡的，这一有趣现象可用于确定一个较好的 ω 值。

图 5.8 曲线表示匹配测试图 5.4 时选用在不同 ω 得到的平均残差随迭代次数的变化曲线。由图看出，ω 选择过大时虽然迭代起始时匹配残差减小很快，但减小到一定值时，迭代开始出现振荡。ω 选择过小，收敛速度变慢。

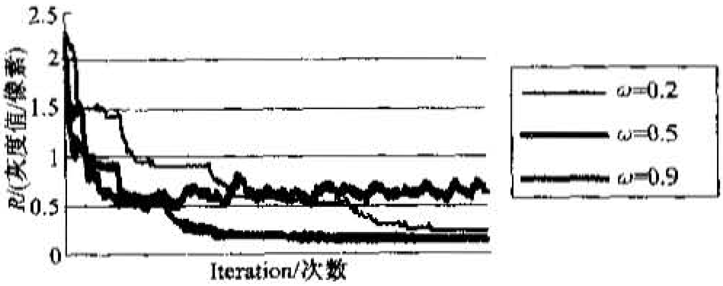


图 5.8　匹配测试图 5.4 在不同 ω 情况下得到的平均残差随迭代次数变化曲线

此外，还可以通过下述公式得到 ω_{opt} 的近似值：

$$\omega_{opt}\approx\frac{2}{1+\sqrt{1-\lambda^2}}。\tag{5.5.1}$$

这里 λ 是矩阵 $\boldsymbol{A}$ 的最大特征值，对于一般图像，如果其长和高的格点间隔都是 h，可得

$$\lambda\approx\frac{1}{2}\left[\cos\frac{\pi}{M}+\cos\frac{\pi}{N}\right]。\tag{5.5.2}$$

这里 M、N 分别是 x 和 y 方向的网格数。

5.6 二维插值和实验验证

为得到亚像元级位移估计，用位移后不位于网格点上的像素的灰度做插值，本文采用最基本的二维线性插值。

5.6.1 二维线性插值

先介绍数值计算中方格网格点上的二维线性插值。对二维情况，假定已知函数值的矩阵 $g[1..m];[1..n]$，另外还已知数组 $x[1..m]$ 及数组 $y[1..n]$。这些输入量的关系由下列函数确定：

$$g[j][k]=g(x[j],y[k])。$$

目的是用插值法计算函数 $g(x,y)$ 在表中未列出的值。

当需要求某点 (x,y)，它是列表中的四个点的包围起来的内部点，为方便起见，采用符合图像常用的坐标系，从右上角按顺时针方向从 1 到 4 给这些点作标记，如图 5.9 所示。更精确地表示如下：如果由

$$x[j]\leqslant x\leqslant x[j+1],$$
$$y[k]\leqslant y\leqslant y[k+1]$$

定义 j 和 k，则

$$g_1=g[j][k],$$
$$g_2=g[j+1][k],$$
$$g_3=g[j+1][k+1],$$
$$g_4=g[j][k+1],$$

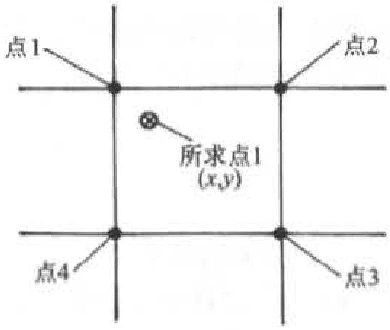


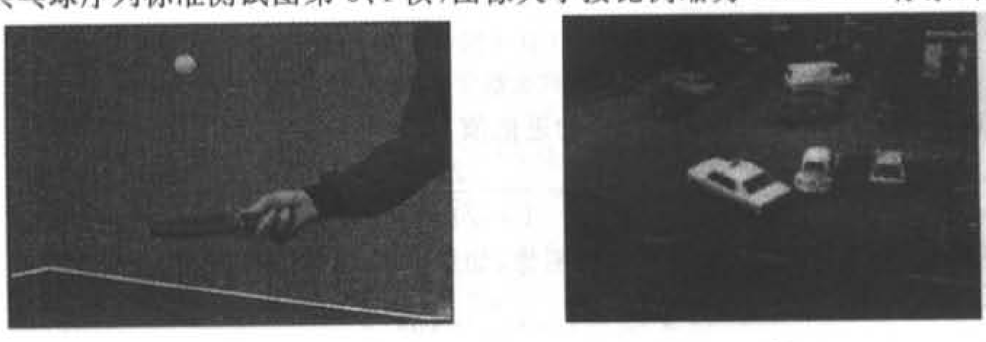
图 5.9 二维线性插值中对点做标记

方格网格点上的二维线性插值公式为

$$t\equiv(x_1-x[j])/(x[j+1]-x[j]),$$
$$u\equiv(y-y[k])/(y[k+1]-y[k]),$$
$$g(x,y)=(1-t)(1-u)g_1+t(1-u)g_2+tug_3+(1-t)ug_4。$$

5.6.2 实验比较采用线性插值前后的结果

为比较用和不用二维线性插值测试图匹配得到的匹配结果，本节用下面的测试图：第 1 组测试图为乒乓球序列标准测试图第 3、4 帧，图像大小按比例缩为 180×122 像素，图 5.10(a)



(a) (b)

图 5.10 标准测试序列

(a) 乒乓球序列标准测试图第 3 帧；(b) 汉堡街头出租车第 0 帧

给出其第3帧。观察序列图像知乒乓球序列图像中有三个运动物体，对第3、4帧图而言分别为：① 向下运动的乒乓球；② 向上运动的手臂及乒乓球拍；③ 向左上角运动的人的腿。第2组测试图为汉堡出租车序列的第0、4帧，图像大小为256×190。图5.10(b)给出汉堡出租车序列的第0帧图像。图像中有四个运动物体：① 转向拐角处的出租车；② 左下角的汽车，从左向右行驶；③ 右下角从右向左的篷车；④ 左上角的很小的一个行人，运动方向为从左向右。

表5.1为比较用和不用二维线性插值测试图匹配得到的残差结果。由表可知用二维线性插值后匹配残差有所减小。

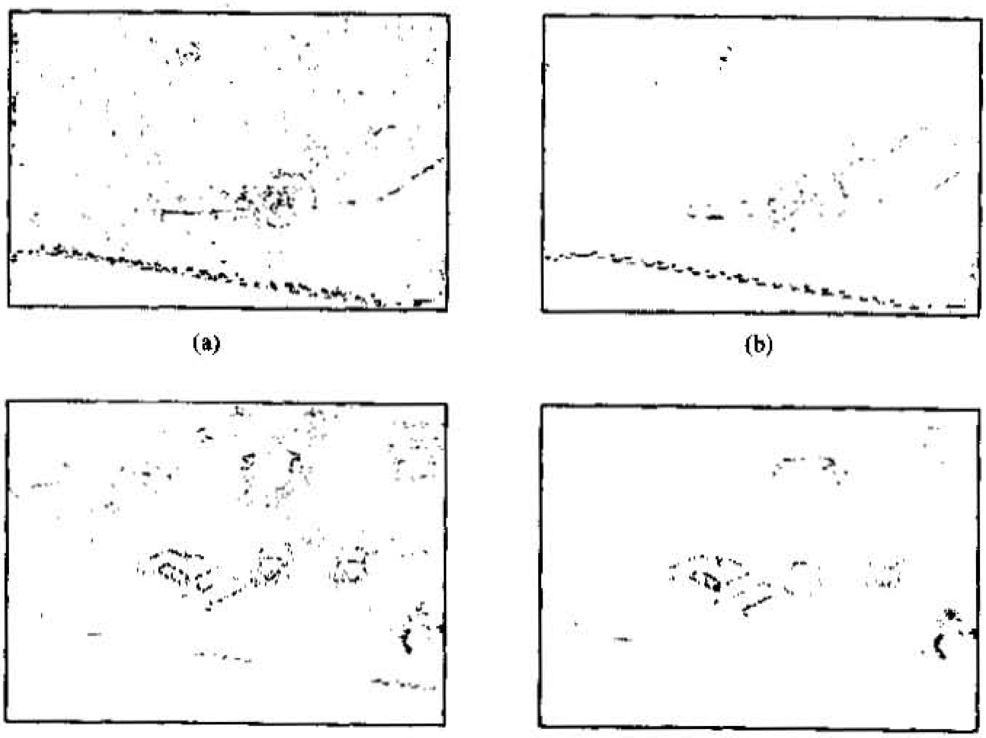
图5.11给出不同测试图像不采用插值算法和采用插值算法的不同匹配残差图；在残差图中点越黑残差越大。可看出采用插值算法匹配残差较小。

图5.12为采用插值算法前后对应情况的位移矢量图。图中小十字代表第一帧图像中的点作为位移起始点，短线表示这一点的位移矢量。由图中可看出采用插值算法后的位移场在静止背景上误运动矢量较少。

实验表明当图像背景有较多纹理，或有多个静止的物体时，用二维线性插值后，无论是残差还是位移场都能得到一定程度的改善。当然，计算工作量也要成倍增加。

表5.1　用二维线性插值前后匹配残差比较

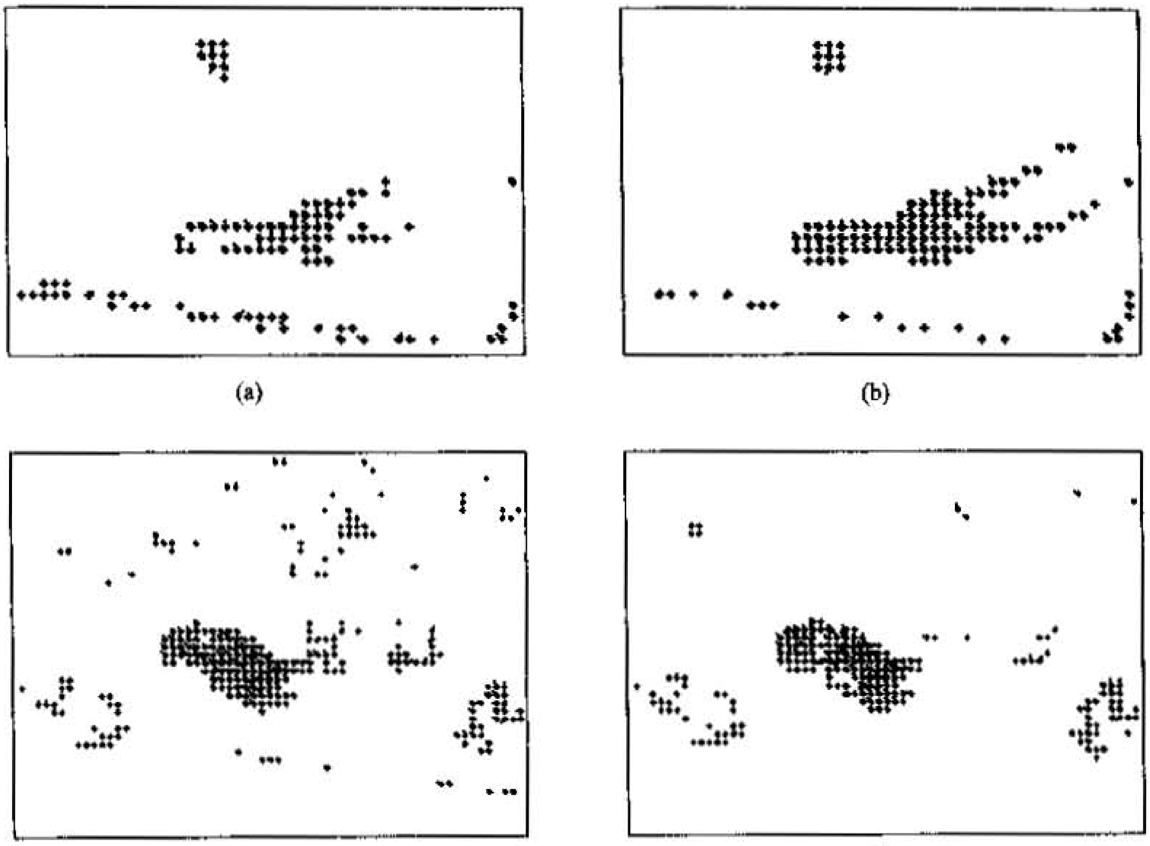
测试图像名	乒乓球测试图		汉堡街头出租车	
参数及是否用二维线性插值	不用插值	用插值	不用插值	用插值
	$\alpha=200.0$ $\omega=0.2$	$\alpha=200.0$ $\omega=0.2$	$\alpha=200.0$ $\omega=0.5$	$\alpha=200.0$ $\omega=0.5$
平均误差 R(灰度值/像素)	3.2362	1.5875	4.0957	2.5614



(a)　(b)　(c)　(d)

图5.11　测试图像采用插值算法前后的匹配残差图比较

(a) 不采用插值算法乒乓球序列第3、4帧图的匹配残差图；(b) 采用插值算法乒乓球序列第3、4帧图的匹配残差图；(c) 不采用插值算法出租车序列第0、4帧图的匹配残差图；(d) 采用插值算法出租车序列第0、4帧图的匹配残差图



(a) (b) (c) (d)

图 5.12 测试图像采用插值算法前后的位移矢量场比较

(a) 不采用插值算法乒乓球序列第 3、4 帧图的位移场；(b) 采用插值算法乒乓球序列第 3、4 帧图的位移场；(c) 不采用插值算法出租车序列第 0、4 帧图的位移场；(d) 采用插值算法出租车序列第 0、4 帧图的位移场

参 考 文 献

[1] Aggarwal J K, Nandhakumar N. On the computation of motion from sequences of images. Proc. IEEE, vol. 76, pp. 917-935, Aug, 1988

[2] Dreschler L and Nagel H H. Volumetric model and 3D trajectory of a moving car derived from nomocular TV frame sequences of a street scene. Computer Graphics and Image Processing, vol. 20, pp. 199-288, 1982

[3] Sclaroff S and Pentland A P. Modal matching for correspondence and recognition. IEEE Trans. Pattern Analysis and Matchine Intelligence, vol. 17, no. 6, pp. 545-561, June 1995

[4] Grimson W E L. Computational experiments with a feature based stereo algorithm. IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-7, no. 1, pp. 17-34, 1985

[5] Jain J R, Jain A K. Displacement measurement and its application in interframe image coding. IEEE Trans. Commun., vol. 29, pp. 1799-1808, 1981

[6] Horn B K P, Schunk B G. Determining Optical Flow. Artificial Intelligence, vol. 17, pp. 185-203, 1981

[7] Gharavi H, Mills M. Block-matching motion estimation algorithms: New results. IEEE Trans. Circ. and Syst., vol. 37, pp. 649-651, 1990

[8] Nagel H H. On the estimation of optical flow: Relations between different approaches and some new results. Artificial Intelligence, vol. 33, pp. 299-324, 1987

[9] Tekalp A M. Digital Video Processing. Prentice Hall PTR, Inc., a Simon & Schuster Company, 1995

[10] 李介谷.计算机视觉的理论和实践.上海:上海交通大学出版社,1991

[11] 马颂德,张正友.计算机视觉.北京:科学出版社,1998

[12] 杨新,郭丰峻等.一种基于梯度算子的图像匹配算法.北京:电子学报,第 27 卷,第 10 期,1999 年 10 月,pp. 30-33

第6章 图像平滑

图像平滑是图像处理中一个重要环节，它对后继处理（如分割）带来很大的影响。图像的平滑和细节保持是一对矛盾关系，图像的低通滤波在降低噪声的同时，产生图像边界的模糊，而人对图像的高频成分是很敏感的。基于偏微分方程的图像平滑方法在这个领域获得了广泛的重视[1]，因为它在平滑噪声的同时，可以使边界得到保持。6.1节介绍热传导方程和线性多尺度空间，热传导方程的解等价于初始图像与高斯滤波器在不同尺度上的卷积。6.2节和6.3节介绍图像的各向异性扩散方程和选择性扩散方程，方程中的扩散系数是由图像的灰度变化决定的。6.4节讲述图像的退化扩散模型，该模型具有明确的几何解释，在沿图像方向上，它进行平滑，而在与边界垂直的方向上不进行任何平滑。此模型对角点具有一定的模糊作用，因此6.5节讨论了对其进行改进的方法。6.6节作了实验验证。

6.1 常系数热传导平滑方程

设初始灰度图像为 $u(x,y,0)$，$u(x,y,t)$为在时间 t 时的平滑图像。则图像的热传导平滑方程为

$$\frac{\partial u(x,y,t)}{\partial t} = \Delta u(x,y,t), \tag{6.1.1a}$$

式中：$\Delta u(x,y,t)$为图像的拉普拉斯算子，其初始条件为 $u(x,y,0)$。方程(6.1.1)的解为

$$u(x,y,t) = G_t * u(x,y,0),$$

这里，$*$ 表示卷积，

$$G_t(x,y) = Ct^{-1}\exp[-(x^2+y^2)/4t] \tag{6.1.1b}$$

是高斯函数。因此，初始图像与不同尺度的高斯滤波器的卷积等价于热传导方程的解[2]。Witkin 采用的尺度空间滤波器就是高斯滤波器。

因果性：当尺度由小变大时，不产生新的细节特征。此特性称为因果性。热传导方程(6.1.1a)具有因果性，随着尺度 t 的增大，图像的平滑效果增强，原始图像中的细节逐渐消失。

定位特性：在不同尺度上，图像特征的空间位置变化情况为定位特性。方程(6.1.1a)在大尺度上所保留下来的图像特征，其空间位置会展宽且发生漂移。

热传导方程的定位特性会变差。为解决这个问题，Witkin 提出，首先在大尺度上检测图像的主要特征，然后尺度逐渐减小，在小尺度上对特征进行跟踪。这样主要特征可获得较为精确的定位。但是，该方法非常复杂，而且不稳定[4]。

由于方程(6.1.1a)所描述的物理过程是各向同性的热扩散现象，因此，其平滑过程自然会带来边界模糊。

6.2 异性扩散方程

针对热扩散方程的缺点，Perona 在上节提到的因果性的基础上给出了图像平滑中多尺度描述的三个准则[5]。

(1) 因果性:同上。

(2) 即时定位:在任一尺度下,图像中的区域边界要明显,并与对应尺度下图像中的物体边界相一致。

(3) 分段平滑:在所有尺度上,区域内部要优先于区域之间进行平滑。

准则(2)表明,在对图像平滑时,图像边界的空间位置不发生变化,在任何尺度上,都可以对边界特征进行精确定位。而准则(3)考虑了图像中的边界条件,在区域边界上,其平滑功能要弱,而在区域内部,其平滑功能要加强。准则(3)为准则(2)提供了方法上的支持。

6.2.1 异性扩散模型

在常系数热传导方程产生的线性尺度空间中,热传导系数为常数1,它没有考虑图像的空间位置。其实,传导系数可根据图像的空间位置来确定。Perona给出的传导系数可变的异性扩散方程如下:

$$\frac{\partial u(x,y,t)}{\partial t} = \mathrm{div}[c(x,y,t)\,\nabla u(x,y,t)]。 \tag{6.2.1}$$

方程(6.2.1)的初始条件为图像$u(x,y,0)$,div为散度算子,∇为梯度算子。当$c(x,y,t)$为常数1时,方程(6.2.1)变为各向同性的热传导方程(6.1.1a)。假设在每瞬时时间(尺度)上,已知图像区域的边界,则在图像区域的内部进行平滑,而在穿过区域边界的位置上,不执行平滑过程。这个过程可通过设置阶跃变化的传导系数来实现,即

$$c(x,y,t) = \begin{cases} 1 & (x,y,t) \in \text{区域内部}, \\ 0 & (x,y,t) \in \text{区域边界}。\end{cases}$$

事实上,区域边界是不可能事先知道的,否则,问题就解决了。因此,需要对相应尺度上的区域边界点进行估计。

设$\boldsymbol{E}(x,y,t)$为边界点的估计,它是一向量函数,$\boldsymbol{E}(x,y,t)=\nabla u(x,y,t)$。即采用梯度算子作为边界点估计。传导系数函数$c(x,y,t)$是以$\|\boldsymbol{E}(x,y,t)\|$为变量的单调下降函数,设

$$c(x,y,t) = g(\|\boldsymbol{E}(x,y,t)\|) = g(\|\nabla u(x,y,t)\|), \tag{6.2.2}$$

则$g(s)=1/\left(1+\frac{\|\nabla I\|}{K}\right)$,$K>0$时,其函数形状见图6.1。

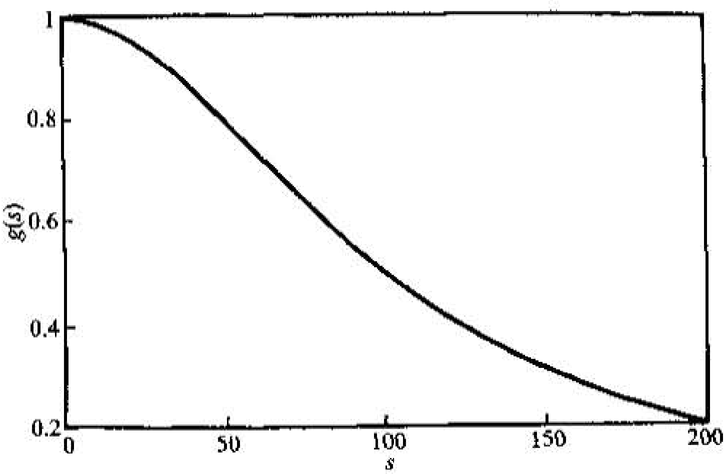


图6.1 传导系数$g(s)$

6.2.2 异性扩散模型的极值原理

设 A 为空间 $\Re^n$ 上一有界开集(即图像中的定义域,一般为矩形区域),$T=(a,b)$为实数轴 R 上一区间。令 D 为空间 $\Re^{n+1}$ 上一开柱体,它由 A 和 T 的叉积形成,即 $D=A\times T=\{(x,t):x\in A,t\in T\}$。设$\partial D$ 为 D 的边界,$\overline{D}$ 为 D 的闭集。$\partial_T D,\partial_S D,\partial_B D$ 分别表示∂D 的上端,侧端和下端。$\partial_T D=\{(x,t):x\in A,t=b\}$,$\partial_S D=\{(x,t):x\in\partial A,t\in T\}$,$\partial_B D=\{(x,t):x\in A,t=a\}$。如图 6.2 所示,其中,$\partial_{SF}D,\partial_{SB}D,\partial_{SL}D,\partial_{SR}D$ 分别表示前侧面,后侧面,左侧面和右侧面,$\partial_S D=\partial_{SF}D\cup\partial_{SB}D\cup\partial_{SL}D\cup\partial_{SR}D$。为了描述方便,令$\partial_{SB}D=\partial_S D\cup\partial_B D$ 表示侧端和底端边界。则有:

极值原理 函数 $f:\Re^{n+1}\to\Re$ 在 $\overline{D}$ 上连续,且在 $D\cup\partial_B D$ 上二次可微,如果 f 在 D 上满足如下微分不等式:$C(x,t)f_t=c(x,t)\Delta f-\nabla c\cdot\nabla f\leqslant 0$,这里,$C:\Re^{n+1}\to\Re^+$ 在 $\overline{D}$ 上连续,并且在 $D\cup\partial_B D$ 上可微,那么,它满足极值原理,即 f 在底端和侧端边界$\partial_{SB}D$ 上取得的极值:$\max\limits_{D}|f|=\max\limits_{\partial_{SB}D}|f|$。

定理的证明见参考文献[6]。

命题 设函数 f 满足上述假设,而且在侧面边界上$\partial_S D$ 上二次可微,且$\nabla_x f=0$,∇_x为沿 x 方向的梯度算子,那么,$\max\limits_{D}|f|=\max\limits_{\partial_B D}|f|$ 命题中的条件 $\nabla_x f=0,x\in\partial_S D$,表示在图像侧端边界上,图像的灰度没有损失,或者说,在侧端边界上,没有能量交换。此条件也称为隔热条件。命题的证明见参考文献[5]。

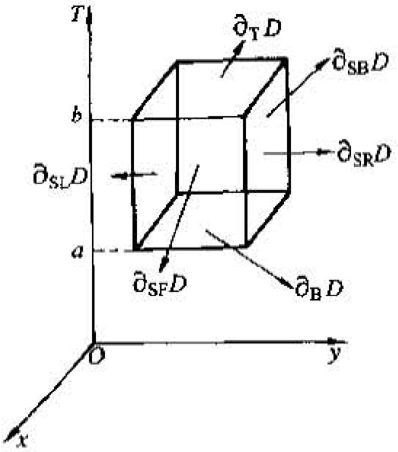


图 6.2 尺度空间中的边界

图像的因果性可通过极值原理来说明:

设图像的细节特征可由图像函数 $u(x,y,0)$的斑点(blob)来描述,那么新斑点的产生表示在图像 $u(x,y,t),t>0$ 中产生了新的极值。方程(6.2.1)是抛物线微分方程的特例,它满足极值原理。由极值原理可看出,图像中的新的极值只能在图像的底端和侧端边界上获得。当添加了隔热条件 $\nabla_x f=0$,$x\in\partial_S D$时,图像中新的极值只能属于底端边界,即初始图像 $u(x,y,0)$。因此,由方程(6.2.1)描述的平滑模型不产生新的细节特征,满足因果性。

6.2.3 边缘增强

采用传统的低通滤波器和上述常系数热扩散方程进行降噪,是以边缘的模糊为代价的[7]。对模糊图像的边缘进行增强可通过高通滤波器或在时间轴上逆向执行扩散方程。但这是一病态问题,计算不稳定,除非对问题进行约束[9]。

对于方程(6.2.1),通过对传导系数进行适当的选择,可使得异性扩散在滤波的同时,进行边缘增强,而且它在时间轴上是正向进行的,通过极值原理,可保证扩散的稳定性。

下面通过一个简单的例子来说明方程(6.2.1)的边缘增强特性。

设图像的边缘为阶跃函数与 Gaussian 函数的卷积。不失一般性,设边缘沿 y 轴方向。因此,散度算子简化为

$$\mathrm{div}[c(x,y,t)\,\nabla u]=\frac{\partial}{\partial x}[c(x,y,t)u_x]。$$

选择传导系数为图像 u 的梯度的函数：$c(x,y,t)=g[u_x(x,y,t)]$，如式(6.2.2)。令 $\phi(u_x)=g(u_x)\cdot u_x$ 表示流量 $c\cdot u_x$。那么，扩散方程在一维情况下变为

$$\frac{\partial u}{\partial t}=\frac{\partial}{\partial x}\phi\left(\frac{\partial u}{\partial x}\right)。$$

图像的边缘增强特性是通过边缘的坡度在尺度空间中的变化来研究的，边缘的梯度可由边缘点的一阶导数 u_x 表示。因此，需要观察 $\partial/\partial t(u_x)$。如果 $c(\cdot)>0$ 且 $u(\cdot)$ 光滑，那么微分的顺序可交换：

$$\frac{\partial}{\partial t}(u_x)=\frac{\partial}{\partial x}(u_t)=\frac{\partial}{\partial x}\left[\frac{\partial}{\partial x}\phi(u_x)\right]=\phi''\cdot u_{xx}^2+\phi'\cdot u_{xxx}。$$

假定边缘的指向为 $u_x>0$。定义边缘点 x 为 $I_{xx}=0$ 且 $I_{xxx}\ll 0$，它对应着最大坡度。如图 6.3 所示。则在边缘点的邻域内，$\partial/\partial t(u_x)$ 的符号与 $\phi'(u_x)$ 相反，如果 $\phi'(u_x)>0$，则边缘的坡度随着时间的增加将降低。相反，若 $\phi'(u_x)<0$，则坡度随时间而增加，从而图像的边缘得到增强。

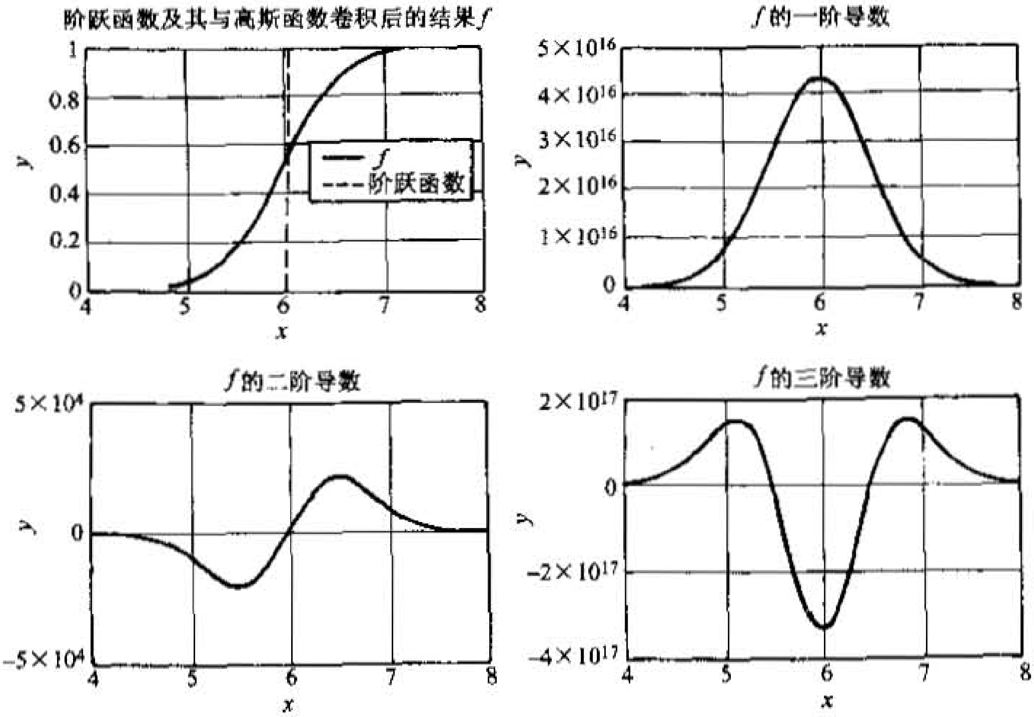


图 6.3　阶跃边缘及其一，二，三阶导数

通过适当选择函数 $\phi(\cdot)$，可获得理想的边缘增强效果。例如：

$$g(u_x)=C/[1+(u_x/K)^{1-\alpha}],\alpha>0,\ C>0。$$

如图 6.4 所示，存在与 K 和 α 相关的门限值 s_t，当 $u_x<s_t$ 时，$\phi(\cdot)$ 单调增加，当 $u_x>s_t$ 时，$\phi(\cdot)$ 单调减小。因此，平滑过程将对图像中灰度变化不大的地方进行平滑，而对灰度变化大的边缘具有增强作用。

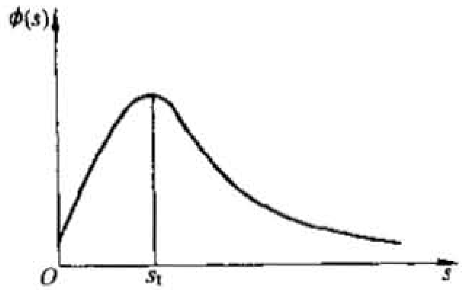


图 6.4　可进行边缘增强的函数 $\phi(\cdot)$ 的形状

6.3 选择平滑方程

采用上述异性扩散模型进行平滑时，图像的边缘在尺度空间可以保持稳定，因此，不需要沿尺度因子逆向跟踪边缘。但是，该模型存在以下两个问题：

(1) 假定图像存在噪声，例如白噪声，则在噪声点(x,y,t)处，图像的梯度$\nabla u(x,y,t)$可能非常大，使得平滑系数$c(x,y,t)$较小，从而将这些噪声点保留下来，降低了去噪性能。

(2) 对于方程(6.2.1)，当扩散系数$g(s)=(1+s^2)^{-1}$或$g(s)=\exp(-s)$时，方程是病态的。实际上，为了使得方程(6.2.1)的解存在且唯一，必须要求$\phi(s)=sg(s)$为非降函数，如果此条件不满足，则此过程是不稳定的。理论上，同一个初始条件会产生多个解[9]。为此，Cattle 提出了选择平滑模型[11]

$$\frac{\partial u(x,y,t)}{\partial t}-\operatorname{div}[g(|G_\sigma\cdot\nabla u|)\nabla u]=0,\tag{6.3.1}$$

其中：

$$u(x,y,0)=u_0(x,y),\quad t\in[0,T]。$$

这里：$G_\sigma(x,y,t)$如式(6.1.1b)。容易看出，它是常系数热方程的基本解。因此，$G_\sigma\cdot\nabla u(x,y,t)$表示常系数热方程解在时刻$\sigma$的梯度。用该项表示图像$u(x,y)$在$(x,y)$的梯度估计。由比方程(6.2.1)和(6.3.1)，可看出，用$|G_\sigma\cdot\nabla u(x,y,t)|$代替方程(6.2.1)中的扩散系数$c(x,y,t)=g(\|\nabla u(x,y,t)\|)$可得到方程(6.3.1)。仅仅这些变动，就避免了方程(6.2.1)上述两方面的不足。$|G_\sigma\cdot\nabla u(x,y,t)|$表示在尺度$\sigma$上对边缘进行估计，然后使用该信息来决定扩散的程度，以避免在边缘点过分扩散。

6.4 退化扩散方程

Cattle 证明了选择平滑模型(6.3.1)的解存在且唯一，但是当尺度因子$\sigma\to0$时，模型并不稳定[10]。另外散度算子括号中包含梯度估计$|G_\sigma\cdot\nabla u|$和梯度∇u，这使得模型没有明确的几何解释。为此，Alvarez 提出了退化扩散模型：

$$\frac{\partial u(x,y,t)}{\partial t}-g(|G_\sigma\cdot\nabla u|)|\nabla u|\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)=0,\tag{6.4.1a}$$

其中：

$$u(x,y,0)=u_0(x,y),\quad t\in[0,T],$$

$u_0(x,y)$为初始的灰度图像；$u(x,y,t)$为在时间t时的图像；G_σ为高斯平滑核；$G_\sigma\cdot\nabla u$为梯度$|\nabla u|$的局部估计；$g(s)$为非增实函数，且$s\to\infty$时，$g(s)$趋向于0。

当$g(s)=1/s$且$\sigma=0$时，得到方程(6.4.1a)的一种极限情况：

$$\frac{\partial u(x,y,t)}{\partial t}=\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right),\tag{6.4.1b}$$

它也是方程(6.2.1)的特例。该极限情况非常重要，因为它的解具有明显的几何解释。实际上，该方程对应下述能量函数的最小化：

$$E(u)=\int|\nabla u(x,y)|\,\mathrm{d}x\mathrm{d}y。$$

设A为具有光滑边界点集合，其特征函数为$u_0=X_A$，那么$E(u_0)$表示集合A的周长。演化方程(6.4.1b)可解释为对特征函数的边界进行平滑。

而方程(6.4.1a)中的因子$|\nabla u|\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$为退化扩散项，在局部坐标系下可化为

$$|\nabla u|\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)=u_{\xi\xi},$$

其中：ξ与梯度∇的方向正交，见图6.5。因此，方程(6.4.1a)表示在方向ξ上进行扩散，而在梯度的方向上不进行任何扩散。在直角坐标系下，$|\nabla u|\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$可分解为：

$$|\nabla u|\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)=\frac{1}{u_x^2+u_y^2}(u_x^2u_{xx}-2u_xu_yu_{xy}+u_y^2u_{yy})。$$

于是，$\frac{\partial u}{\partial t}=|\nabla u|\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$的解的水平集还可解释为以平均曲率沿法线方向进行运动。该方程在图像的分割中具有广泛的应用。

方程(6.4.1a)中的因子$g(|G_\sigma\cdot\nabla u|)$用于边缘的增强。它控制着扩散的速度。因为$G_\sigma\cdot\nabla u$表示梯度图像的高斯平滑，如果梯度图像在点$(x,y)$处的邻域内具有较小的加权平均值，则点$(x,y)$领域内的灰度变化比较平缓。因为$g(s)$为非增函数，所以此处的扩散速度比较强。反之，在图像的边缘点上，其梯度图像的邻域内具有较大的加权平均值，从而扩散速度较小。所以，退化扩散模型的解使得图像边缘得以保持，而灰度变化不大的地方更加光滑。

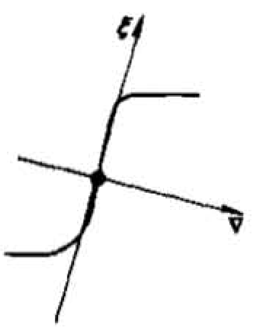


图6.5　边缘点及局部坐标

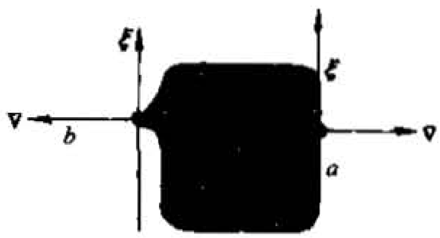


图6.6　轮廓线上的线条点和角点的局部坐标

当(x,y)邻域内的灰度变化较小时，上述扩散速度虽然较大，但它仍然仅是沿着与梯度正交的方向进行扩散。这样的扩散效率是很低的。在这种情况下，既然(x,y)邻域内不包含图像信息(灰度变化平缓)，完全可以采用各向同性的热扩散方程方法。综合上述情况，得到下述非线性扩散方程：

$$\frac{\partial u}{\partial t}-g(|G_\sigma\cdot\nabla u|)\left((1-h(|\nabla u|))\Delta u+h(|\nabla u|)|\nabla u|\operatorname{div}\frac{\nabla u}{|\nabla u|}\right)=0,$$

$$u(x,y,0)=u_0(x,y)\quad t\in[0,T],\tag{6.4.2}$$

这里：$h(s)$为光滑非降函数，当$s>aE_0$时，$h(s)=1$，此时执行的是退化扩散方程，它在与梯度正交的方向上扩散；当$s<E_0$时，$h(s)=0$，此时模型为热扩散方程，它在各个方向上进行相同的扩散，这对于去除孤立噪声是很有作用的。这里，a为大于1的常数，E_0为常数。Alvarez证明了方程(6.4.2)是适定的。

6.5 保持特征点的扩散模型

6.5.1 数学模型

对图像进行平滑时，应该是在去噪声的同时，保持信号的特征。为此，平滑应该满足如下两个原则：

(1) 平滑速度。在具有图像特征的位置上，平滑速度应该降低。在没有明显特征的地方，平滑速度应该增加。

(2) 平滑方向。在穿过图像特征的方向上，不应该进行平滑，否则将使得特征模糊化。在沿图像特征的方向上，可进行平滑。

在方程(6.4.1a)中，平滑速度所使用的信息仅为梯度。具体来说，当 $h(s)=1$ 时，退化扩散方程沿与梯度正交的方向进行扩散。其中 $g(s)$ 为扩散速度，ξ 为扩散方向。对于一维特征来讲，如线条边缘，此扩散方法可以保持图像的特征。如图 6.6 中的点 a，与梯度正交的方向即为线条的方向。沿线条方向进行扩散时，不会造成点 a 的模糊化。梯度的模表示灰度变化的快慢，由于扩散速度为非增函数，从而由其决定的扩散速度也是合理的。但是，对于二维特征来讲，如角点，仅靠梯度信息来决定扩散速度及方向显然是不够的。如图 6.6 中的点 b，若沿与梯度正交的方向 ξ 进行扩散，则点 b 的灰度必将被轮廓线外的、ξ 方向上的点的灰度模糊化。而且，随迭代次数的增加，角点 b 的信息将丧失掉。所以，对于角点，不能沿与梯度正交的方向进行扩散。

在本节的扩散方案中，扩散的方向与退化扩散模型是相同的，但是，扩散速度是不同的。在线条点上，扩散速度和退化扩散模型是相同的。在角点上，扩散速度将趋于 0。这样，角点灰度将不会随迭代的增加而发生改变。因此，关键是能够将线条点和角点进行区分开来。为此，设 $G=G_\sigma * u$ 为高斯平滑后的图像，$|G_\nabla|=\sqrt{G_x^2+G_y^2}$ 为高斯平滑图像的梯度模。ξ 与 ∇ 正交。为说明问题，假设一维函数为 $u(x)$，经过等间隔采样，采样间隔为 h，得到 N 个离散点 u_0，$u_1,\cdots,u_i,\cdots,u_j,\cdots,u_N$，设当前点为 u_i，求解 u_i 的一阶导数和二级导数。通过中心差分来逼近其一阶导数，即

$$u'_i=(u_{i+0.5}-u_{i-0.5})/h。$$

用同样的方式可求出 $u'_{i+0.5}$ 和 $u'_{i-0.5}$，u_i 的二阶导数为

$$u''_i=(u'_{i+0.5}-u'_{i-0.5})/h^2=(u_{i+1}+u_{i-1}-2u_i)/h^2。$$

二阶方向导数反映了两侧点与当前点的差异情况。我们以二阶方向导数 $G_{\xi\xi}$ 来区分角点和线条点。对于线条上的点来说，ξ 的方向和线条方向是相同的，因此，其中心点与两侧点的灰度差异不大。而对于角点来说，在方向 ξ 上，其中心点与两侧点的灰度差异比较大。这可以从图 6.6 中点 a 和点 b 反映出来。所以，当 $|G_{\xi\xi}|<A$ 时，点 (x,y) 为线条点。当 $|G_{\xi\xi}|>B$ 时，点 (x,y) 为角点。$0<A<B$。我们将使用梯度信息和方向 ξ 上的二阶导数 $G_{\xi\xi}$ 来共同决定扩散速度。具体来说，将原来的扩散速度 $g(|G_\nabla|)$ 改进为

$$g(|G_\nabla|,|G_{\xi\xi}|)=\frac{1}{1+|G_\nabla|^2+T(|G_{\xi\xi}|)}。\tag{6.5.1}$$

Alvarez 指出，这种类型的方程适定的假设条件之一为：函数 $g(X)\in C^{1,1}(R^2,R)$ 且 $X\in R^2$，$g(X)>0$。$C^{1,1}(R^2,R)$ 表示所有一阶导数为 Lipschitz 连续函数构成的空间。为此，可采用函数逼近的方法来保证这个条件。具体来说，设 $1>\varepsilon_1,\varepsilon_2>0$，当 $|G_{\xi\xi}|>Th+\varepsilon_2$ 时，$T(|G_{\xi\xi}|)=1/\varepsilon_1$。当 $|G_{\xi\xi}|<Th-\varepsilon_2$ 时，$T(|G_{\xi\xi}|)=\varepsilon_1$。在 $|G_{\xi\xi}|=Th$ 两侧 ε_2 邻域内进行光滑逼近。为了保证扩散速

度的一阶导数 Lipschitz 连续，只要 $T(|G_{\xi\xi}|)$二阶可导即可。因此，在区间$[Th-\varepsilon_2, Th+\varepsilon_2]$使用三次多项式逼近便可满足二阶可导。当 $\varepsilon_1, \varepsilon_2 \to 0$ 时，$T(|G_{\xi\xi}|)$为幅值 $1/\varepsilon_1$ 的负脉冲函数。

最后得到改进的扩散方程为

$$\frac{\partial u}{\partial t} - g(|G_\nabla|, |G_{\xi\xi}|)((1-h(|\nabla u|))\Delta u + h(|\nabla u|)|\nabla u| \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) = 0,$$

$$u(x,y,0) = u_0(x,y), \quad t \in [0,T]。\tag{6.5.2}$$

6.5.2 数值计算方法

(1) 扩散算子的逼近。为了计算退化扩散算子$|\nabla u| \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$，必须在离散网格上进行逼近。设当前点为$(i,j)$，网格的角度及四个方向如下(见图 6.7)：

θ_0 为 0：$(i,-1,j) \to (i+1,j)$，　θ_1 为 $\pi/4$：$(i,-1,j-1) \to (i+1,j+1)$，

θ_2 为 $\pi/2$：$(i,j-1) \to (i,j+1)$，　θ_3 为 $3\pi/4$：$(i,+1,j-1) \to (i-1,j+1)$。

以上述四个方向二阶导数的线性组合来逼近 $u_{\xi\xi} = |\nabla u| \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)$，即 $|\nabla u| \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) = \sum_{n=0}^{3} f(\theta-\theta_n)(\partial^2 u^{k+1}/\partial x_n^2)$。其中：$\theta$ 为 ξ 方向的角度，它与梯度垂直。系数函数 $f(\cdot)$的定义域为$\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$，它是偶函数。

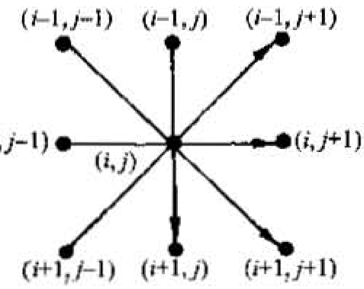


图 6.7　网格方向

$$f(x) = \begin{cases} 1, & |x| \leqslant \theta_t; \\ 0, & |x| \geqslant \frac{\pi}{4} - \theta_t, \theta_t \text{ 为小角度常数}; \\ \left(\frac{4}{\pi}\right)^4 \left(|x| - \frac{\pi}{4} + \theta_t\right)^4 & \text{其他。} \end{cases}$$

$\partial^2 u^{k+1}/\partial x_n^2$ 为上述四个方向的二阶导数，其差分格式为

$$\partial^2 u^{k+1}/\partial x_n^2 = (m^2+1^2)^{-1}[u^{k+1}(i+m,j+1) - 2u^{k+1}(i,j) + u^{k+1}(i-m,j-1)]。$$

(2) 离散化。梯度算子∇u 的离散化：对于 3×3 模板，方向导数 u_x 和 u_y 的模板系数分别见表 6.1 和表 6.2。其中，$A=(2^{\frac{1}{2}}-1)/(2-2^{\frac{1}{2}})$，$B=(2+4\cdot A)^{-1}$，$C=A\cdot B$。

表 6.1　u_x 方向的模板系数

$-C$	$-B$	$-C$
0	0	0
C	B	C

表 6.2　u_y 方向的模板系数

$-C$	0	C
$-B$	0	B
$-C$	0	C

拉普拉斯算子 Δu 的离散化：

$$\begin{aligned}\Delta u^{k+1}(i,j) = &0.5[u^{k+1}(i+1,j) + u^{k+1}(i-1,j) + u^{k+1}(i,j+1) + u^{k+1}(i,j-1)] \\ &+ 0.25[u^{k+1}(i+1,j+1) + u^{k+1}(i-1,j-1) \\ &+ u^{k+1}(i+1,j-1) + u^{k+1}(i-1,j+1)] - 3u^{k+1}(i,j)\end{aligned}$$

u_t 的离散化：$u_t = \frac{u^{k+1}-u^k}{\delta_t}$，$\delta_t$ 为采样时间轴上的采样间隔。

(3) 方程组的求解。经过上述离散化过程，方程(6.5.2)离散化为

$$\frac{u^{k+1}-u^k}{\delta t} = g(|G_\nabla|, |G_{\xi\xi}|)[(1-h(|\nabla u^k|)]\Delta u^{k+1} + h(|\nabla u^k|)\sum_{n=0}^{3} f(\theta-\theta_n)\left(\frac{\partial^2 u^{k+1}}{\partial x_n^2}\right)。\tag{6.5.3}$$

由 u^k 求 u^{k+1} 的过程中，将 $(k+1)\delta_t$ 时刻图像 $u^{k+1}(i,j), i=1,2,\cdots,N,\ j=1,2,\cdots,M$ 按列从第一列到最后一列进行排列，得到 $(N\times M)\times 1$ 列向量，设其为 $\boldsymbol{x}$。其中：N 为图像的高度，M 为图像的宽度。设系数矩阵为 $\boldsymbol{A}$，则 $\boldsymbol{A}$ 为 $(N\times M)\times(N\times M)$ 矩阵。在 $(k+1)\delta_t$ 时刻，$u^{k+1}(i,j)$ 的数值只与其邻域的像素有关，因此，矩阵 $\boldsymbol{A}$ 为大型稀疏矩阵。其分块矩阵形式为：

$$
\begin{bmatrix}
\boldsymbol{A}_1 & \boldsymbol{B}_1 & 0 & 0 & 0 & 0 & 0 \\
\boldsymbol{C}_1 & \boldsymbol{A}_2 & \boldsymbol{B}_2 & 0 & 0 & 0 & 0 \\
0 & \boldsymbol{C}_2 & \boldsymbol{A}_3 & \boldsymbol{B}_3 & 0 & 0 & 0 \\
\vdots & & & & & & \vdots \\
0 & 0 & 0 & \boldsymbol{C}_{i-1} & \boldsymbol{A}_i & \boldsymbol{B}_i & 0 \\
\vdots & & & & & & \vdots \\
0 & 0 & 0 & 0 & 0 & \boldsymbol{C}_{M-1} & \boldsymbol{A}_M
\end{bmatrix}
$$

其中每一分块矩阵的大小为 $N\times M$，而且每行中，它最多有三个元素不为 0。设图像为 256×256，数据类型为 double(8 个字节)，则矩阵 $\boldsymbol{A}$ 的存储空间需要(256×256)×(256×256)×8，大约为 $32\mathrm{K}^3$。因此，需要对矩阵 **A** 进行压缩。由于 **A** 中的每行最多有 9 个元素不为 0，因此，每行仅分配 9×8 个字节。这样，矩阵 **A** 的大小仅为 4.5M。常数矩阵 $\boldsymbol{B}$ 由 $(k)\delta_t$ 时刻的图像 u^k 形成。最后，得到线性方程组：

$$\boldsymbol{AX}=\boldsymbol{B}。$$

采用收敛散度较快的超松弛迭代算法进行求解：

$$x_i^{k+1}=x_i^k+\frac{\omega}{a_{i,j}}\Big[-\sum_{j=0}^{i-1}a_{i,j}x_j^{k+1}-\sum_{j=i}^{n}a_{i,j}x_j^k+b_i\Big],$$

其中：$i=1,2,\cdots,n$，为变量下标；$k=0,1,2,\cdots$ 为迭代步数；$0<\omega<1$ 为松弛因子。

(4) 计算流程见图 6.8。

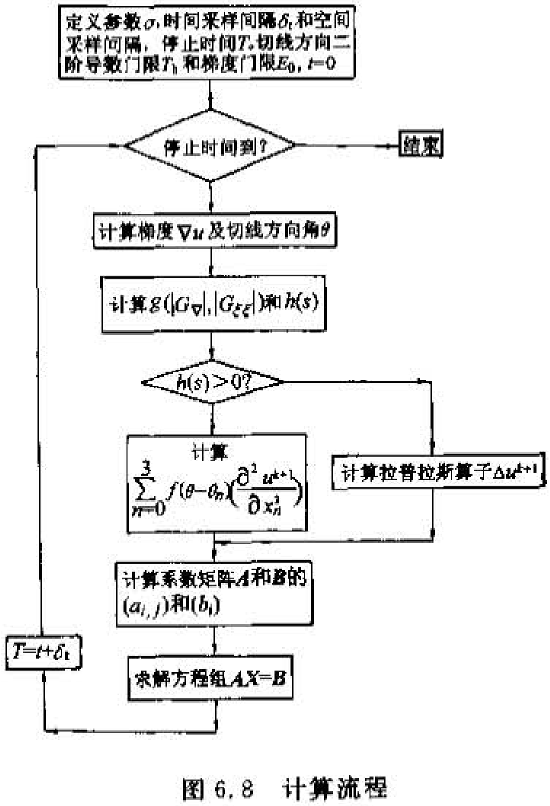


图 6.8　计算流程

6.6　实验结果和分析

指纹作为一种独特的个人生理特征，越来越受到广泛重视。许多场合，如刑事证据，在线身份验证，都需要对指纹进行比对，这就需要有指纹分类、特征提取、匹配等技术。但是，获取的指纹图像往往可能受到各种噪声源的干扰，如传感器噪声、指纹本身受到污染等，影响到图像的后续处理。所以有必要对获取的指纹图像进行适当的降噪等预处理。但是，在平滑降噪领域一直存在一个棘手的问题，即平滑和保边界保细节的矛盾。本文采用了基于偏微分方程的图像平滑方法对指纹图像进行降噪处理，并给出了比较结果。在计算 Alvarez 模型和第6.5节改进算法的过程中，取相同的参数：演化时间 t 为 5，高斯窗 $\sigma=1$，退化扩散门限 $E_0=2$，$A=1$，曲率门限 $T_h=4$。

显然，采用 3×3 模板的中值滤波结果（见图 6.9(b)）中，残留了相当大的背景噪声。同时，如果维纳滤波算法中的局部邻域大小取 3×3，结果也同样是残留了很大的噪声。但是，如果这两种算法采用 5×5 的模板，则结果如图 6.9 中的(c)和(d)所示，纹理被模糊了。Alvarez 模型的平滑结果中，背景噪声几乎全部滤去，比较平直的线条也保留得很好，但是很多曲率较大的边缘点被模糊了，而改进算法的结果则显然兼顾了降噪和保留边界，效果比较令人满意。

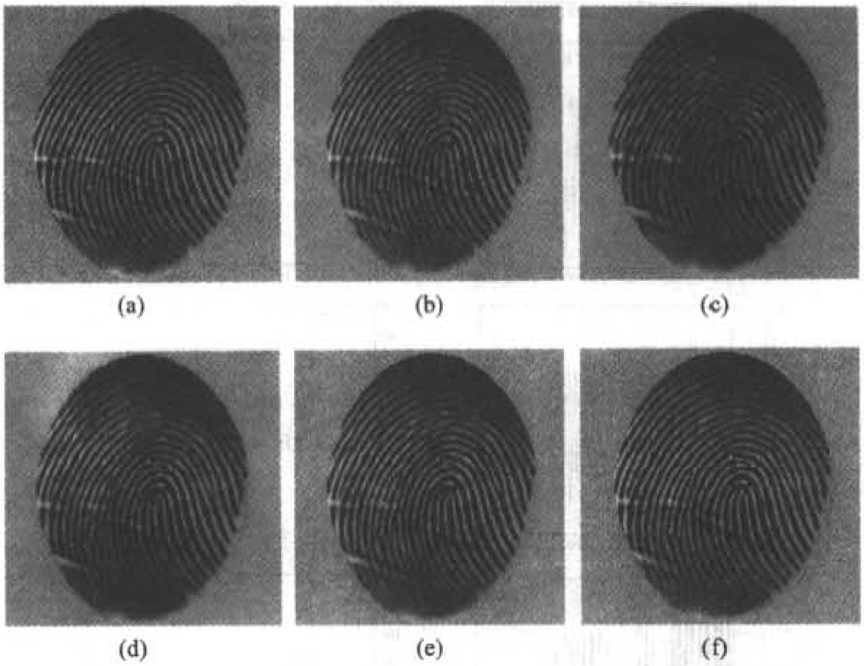


图 6.9　几种指纹预处理结果的比较

(a)带噪指纹原图；(b) 采用 3×3 模板的中值滤波结果；

(c) 采用 5×5 模板的中值滤波结果；(d) 维纳滤波结果，其中局部邻域大小为 5×5；

(e) Alvarez 模型的平滑结果；(f) 改进算法平滑结果

对于指纹分类、指纹识别等的处理，希望指纹图像中的脊清晰，与两侧的谷部分有较大的灰度差（见图 6.10(a)）。如果纹理被模糊了，则灰度差减小（见图 6.10(b)），或者有随机点噪声的干扰，则灰度图上产生毛刺（见图 6.10(c)），不利于正确的识别。

为了检验各种算法的降噪效果，图 6.11 的(a)组、(b)组分别对应图 6.9 中 y 坐标 128 和 220 处的灰度变化曲线。显然，在 y 为 128 的横截线上，由于没有大曲率点，纹理本身也很清晰，所以几种滤波结果相差不很悬殊，当然还是容易看出有差异，比如在 x 坐标为 150 的地方，

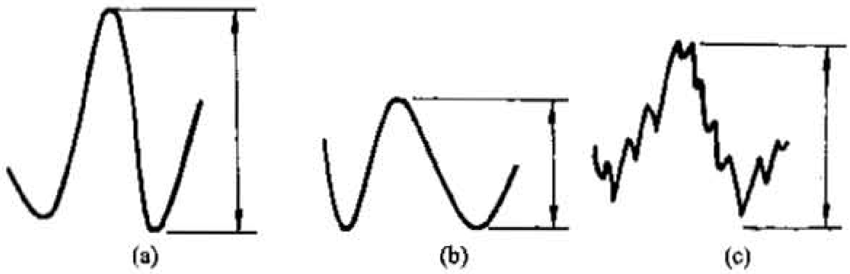


图 6.10　指纹横截线上局部灰度变化示意图

在 y 为 220 的横截线上，很明显看出改进算法的效果。首先是曲线的中间部分（对应于原指纹图像的纹理部分），几种对比算法的结果中，波峰波谷很明显被削掉了很多，其次左右两侧（对应于原图的背景部分）仍然存在较多的背景噪声。

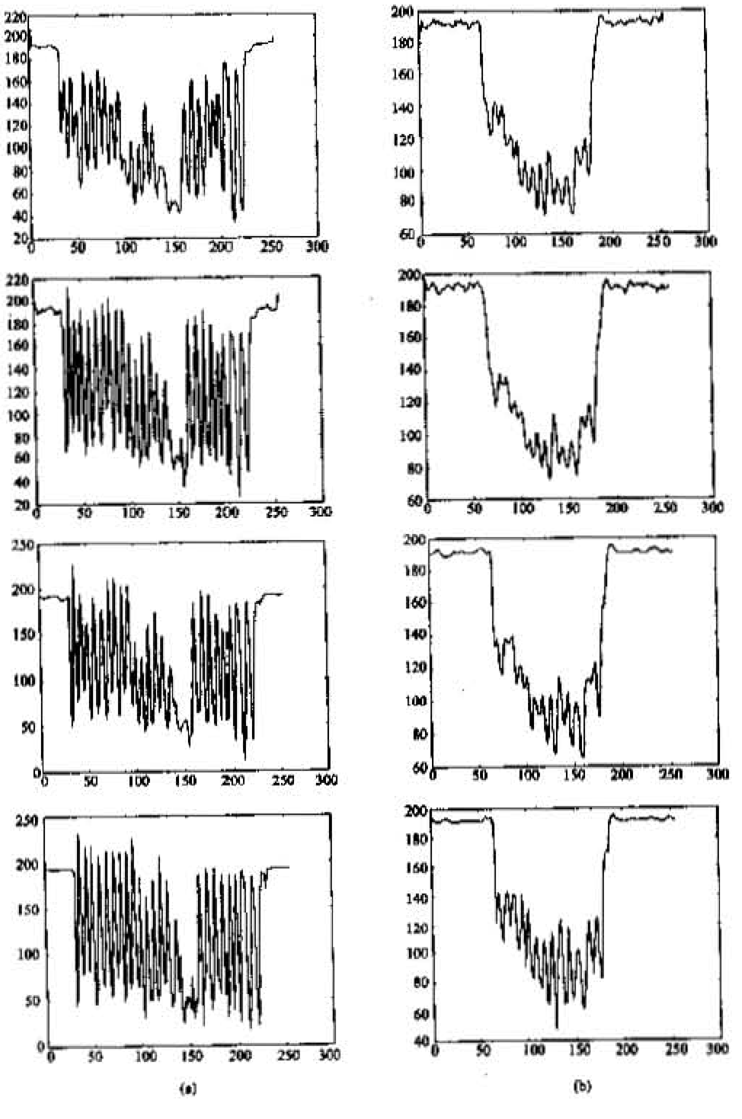


图 6.11　指纹图横截线上灰度变化曲线

(a)组对应 $y=128$；(b) 组对应 $y=220$，从上到下依次对应以下四种计算结果：采用 5×5 模板的中值滤波结果；维纳滤波结果，其中局部邻域大小为 5×5；Alvarez 模型的计算结果；改进算法的计算结果

参 考 文 献

[1] Sapiro G. Geometric partial differential equations and image analysis. Cambridge: Cambridge University Press, 2001

[2] Koenderink J J. The structure of images. Biol. Cybernet., 50(1984), pp. 363-370

[3] Witkin A. Scale-space filtering. Int. Joint Conf. Artificial Intelligence, Karlsruhe, West Germany, 1983, pp. 1019-1021

[4] Clare A. Singularity theory and phantom edges in scale space. IEEE Trans. Pattern Anal. Machine Intell., vol. 10. no. 5, pp. 720-727. 1988

[5] Perona P, Malik J. Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Machine Intell., vol. 12. no. 7, pp. 720-727. July 1990

[6] Nirenbarg L. A strong maximum principle for parabolic equations. Commun. Pure Appl. Math., vol. VI, pp. 167-177, 1953

[7] Canny J. A computational approach to edge detection. IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-8, pp. 679-698, 1986

[8] Hummel A, Kimia B, Zucker S. De-blurring Gaussian blur. Comput. Vision, Graphics, Image Processing, vol. 38, pp. 66-80, 1987

[9] Holling K, Nohel J A. A diffusion equation with a monmonotone constitutive function. Proc. On Systems of Nonlinear Partial Differential Equations, John Ball; ed., Reidel, Boston, MA, 1983, pp. 409-422

[10] Alvarez L, Lions P L, Morel J M. Image selective smoothing and edge detection by nonlinear diffusion Ⅱ. SAIM Number. Anal., vol. 29, no. 3, pp. 845-866, June 1992

[11] Catte F, Coll T, Lions P L, et al. Image selective smoothing and edge detection by nonlinear diffusion. SIAM J. Numer. Anal., 29(1992), pp. 182-193

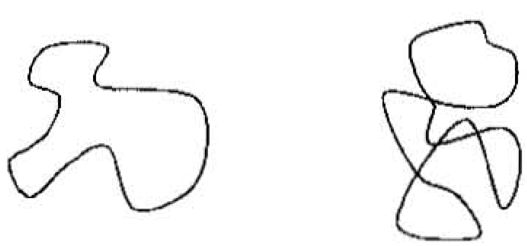
第7章 几何曲线演化和水平集方法

水平集方法是一种新颖的求解几何曲线演化的方法，它以一种隐含的方式来表达平面闭合曲线或者立体闭合曲面，从而避免了对闭合曲线演化过程中的跟踪，将曲线演化转化成一个纯粹的求偏微分方程数值解问题。另外，如前所述，水平集方法应用于几何曲线演化时，避免了演化曲线的参数化过程，这样，曲线的拓扑结构的变化，即分裂或合并，就变得非常自然，这种特点是基于参数的曲线变形模型所不具有的。因此，近年来，水平集方法逐渐成为研究的热点，并被应用于图像分割[1]、图像平滑[2]、运动分割以及运动目标跟踪[3][4]，甚至立体视觉[5]和图像的修复[6]。本章将从几个方面讨论水平集方法。首先，介绍几何曲线演化理论，作为基于几何曲线演化模型图像分割的基础，并引出水平集方法；其次，详细介绍水平集理论，并给出水平集方法的数值计算方法；为了使得水平集方法能更好地应用于图像处理的各个领域，本章还讨论了水平集函数演化的快速方法，以及在实现水平集方法时的一些特殊问题，包括符号距离函数的快速计算，扩展速度场的构造，以及演化时间间隔的选择等；本章还讨论了几种加快水平集方法计算速度并提高计算稳定性的方法，并给出了实验比较。另外，本章将阐述基于Mumford-Shah 模型的图像分割方法，并提出一种改进的具有全局优化特点的活动轮廓线分割方法。

7.1 曲线演化理论和图像分割

7.1.1 曲线演化理论

本章所研究的曲线演化对象仅限于简单闭合曲线，即图 7.1(a)所示没有交叉的曲线。



(a) (b)

图 7.1 闭合曲线类型

(a) 简单闭合曲线；(b) 非简单闭合曲线

描述曲线几何特征的两个重要参数是单位法矢和曲率，单位法矢描述曲线的方向，而曲率则描述曲线弯曲的程度。曲线演化理论就是仅利用曲线的单位法矢和曲率等几何参数来研究曲线随时间的变形，而在参数曲线演化模型中，曲线演化依赖的是任意参数化曲线的导数。假设演化曲线为 $C(p,t)=[x(p,t),y(p,t)]$，这里的 p 是任意参数化变量，而 t 是时间。设曲线的内向单位法向量为 $\boldsymbol{N}$，曲率为 k，则曲线沿其单位法矢方向的演化过程可以用如下偏微分方程表示：

$$\frac{\partial C}{\partial t}=V(k)\boldsymbol{N} \tag{7.1.1}$$

上式中的 $V(k)$ 是速度函数，决定曲线上每点的演化速度，如图 7.2 中的箭头所示为曲线的单位法矢，曲线的演化就沿着单位法矢的方向。要指出的是，沿任意方向运动的曲线总是可以重

新参数化为方程(7.1.1)的形式[7]，因为曲线的切线方向的变形仅仅影响曲线的参数化，不会改变其形状和几何属性。

研究中最常用的是曲率演化和常量演化。曲率演化由如下几何热力学方程描述：

$$\frac{\partial C}{\partial t} = \alpha k \boldsymbol{N} \tag{7.1.2}$$

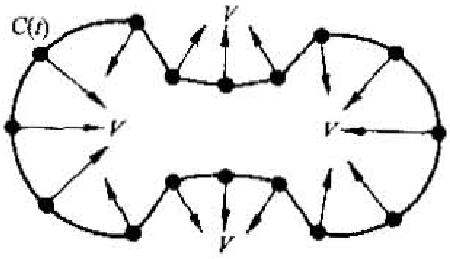


图 7.2　曲线演化示意

式中：α 是正值常数；k 是曲线的曲率，可以证明[8]，任意形状的简单闭合曲线，在上述偏微分方程的驱动下，将会逐渐变平滑，并逐渐收缩成一个圆点，如图 7.3 所示。图 7.3(a)是初始闭合曲线，图 7.3(b)、(c)、(d)是图(a)的曲线在曲率演化方程下的变形过程，曲线逐渐收缩平滑，并最后收缩成一个点。曲率演化与参数化变形模型中的弹性内力的变形效果非常相似。

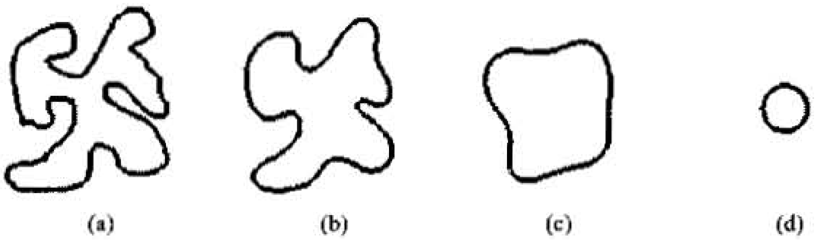


图 7.3　曲线的曲率演化过程

常量演化方程如下式所示：

$$\frac{\partial C}{\partial t} = V_0 \boldsymbol{N} \tag{7.1.3}$$

上式中的 V_0 是一个决定曲线演化速度和方向的常系数。常量演化和参数变形模型中的气压力对曲线具有相同的变形效果。图 7.4 就是曲线的常量演化过程(收缩演化)。图 7.4(a)是原始曲线，图 7.4(b)、(c)、(d)是曲线的常量演化阶段图像，可以看到，常量演化会导致曲线出现尖角，并可能出现拓扑结构的变化(分裂或合并)。

曲率变形与常量变形的变形特性刚好相反，曲率变形消除曲线的角点并使曲线变光滑，而常数变形则使曲线产生角点。

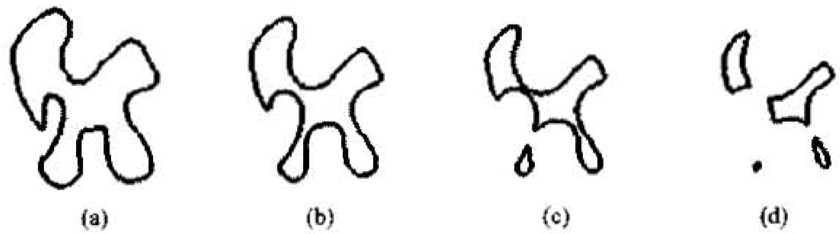


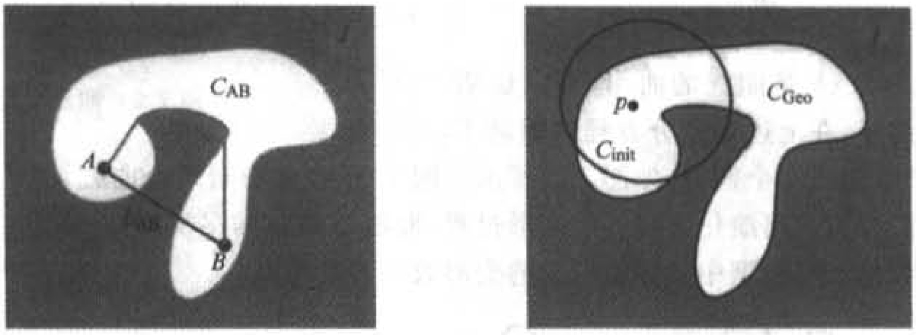
图 7.4　闭合曲线的常量演化过程($V_0=1$)

7.1.2　几何曲线变形和图像分割

本节讨论如何将几何曲线变形模型应用于图像的分割中。为此，先介绍测地轮廓线的概念[9]。

考察图 7.5(a)中位于图像 I 中的 A、B 两点之间的距离定义。我们知道，点 A 和 B 之间的最短欧氏距离是线段 L_{AB} 的长度，即 A、B 两点之间的直线距离。然而，假设我们认为图像 I

中的灰色区域是高原，而白色区域是平原，如果一个从 A 走到 B，则 A 和 B 之间的最短长度就不再是 L_{AB} 的长度了，而是图中曲线 GC_{AB} 的长度。显然，如果我们考虑图像的某种特征作为图像中两点之间长度的加权，则这两点之间的最短距离将不再是欧氏距离。这里的距离，引用测地学术语，称为测地距离。更精确的数学定义如下。



(a) (b)

图 7.5 测地距离和边界检测

(a) 几何距离和测地距离；(b) 轮廓线的几何最短长度和测地最短长度

设 $g(I)$ 为定义于图像 $I:[0,a]\times[0,b]\rightarrow\Re^+$ 中的某种属性函数，A、B 是图像 I 中任意两点，$C_{AB}(p):[0,1]\rightarrow\Re^2$ 为连接 A 和 B 点之间的任意连续曲线，则 C_{AB} 的欧氏几何长度

$$L(C_{AB}) = \int_{C_{AB}} \mathrm{d}s = \int_0^1 |C'_{AB}(p)|\,\mathrm{d}p, \tag{7.1.4}$$

上式中的 $\mathrm{d}s$ 是欧氏弧长微元。则定义 C_{AB} 关于属性函数 $g(I)$ 的测地距离[9]

$$GL(C_{AB}) = \int_{C_{AB}} g[I(C_{AB})]\mathrm{d}s = \int_0^1 g[I(C_{AB})]\,|C'_{AB}(p)|\,\mathrm{d}p, \tag{7.1.5}$$

则定义点 A 和 B 之间的测地距离是满足

$$GC_{AB} = \arg\min_{C_{AB}} GL(C_{AB}) \tag{7.1.6}$$

的连接 A、B 两点的曲线 GC_{AB} 的长度。显然，GC_{AB} 是连接 A 和 B 两点中测地长度最短的曲线。

按照上述定义，针对图 7.5(a)，设 $g(I)=1/(\varepsilon+I)^2$，$\varepsilon>0$ 为一个小值，则点 A 和 B 之间的测地距离就是图中曲线 GC_{AB} 的测地长度。

下面我们考虑闭合曲线 $C(p)$（比如图 7.5(b)中的曲线 C_{init}）试图分别按照减小其几何长度和基于 $g(I)$ 的测地长度的方式缩短，考虑 $C(p)$ 变化的形式。则此时 $C(p)$ 的几何长度

$$L(C) = \oint_C \mathrm{d}s = \oint |C'(p)|\,\mathrm{d}p, \tag{7.1.7}$$

基于 $g(I)$ 的测地长度

$$GL(C) = \oint_C g[I(C)]\mathrm{d}s = \oint g[I(C)]\,|C'(p)|\,\mathrm{d}p。 \tag{7.1.8}$$

$C(p)$ 按照两种长度度量方式缩短，也就是最小化式(7.1.7)和(7.1.8)，利用变分法，可得上两式的欧拉-拉格朗日方程如下[9]：

$$\frac{\partial C}{\partial t} = k\boldsymbol{N}, \tag{7.1.9}$$

$$\frac{\partial C}{\partial t} = \{g[(I(C))]k - \nabla g[I(C)]\cdot \boldsymbol{N}\}\boldsymbol{N}, \tag{7.1.10}$$

式中：k 为 $C(p)$ 的曲率；N 为内向单位法矢。

遵照方程(7.1.9)，闭合曲线 $C(p)$ 将以最快的速度缩短其欧氏几何长度 $L(C)$，比较式(7.1.9)和(7.1.2)可知，式(7.1.9)其实就是几何曲率变形方程，曲线 $C(p)$ 将逐渐收缩成一个点。图7.5(b)中的点 p 就是 C_{init} 缩短到最短几何长度的形状。

而遵照方程(7.1.10)，$C(p)$ 将以最快的速度缩短其基于属性函数 $g(I)$ 的测地长度。然而，$C(p)$ 缩短到最小测地长度时的形状，就不一定是一个点了，而和属性函数 $g(I)$ 的选择有直接的关系。由于 $g(I)$ 是图像数据的函数，因此，我们可以设计 $g(I)$ 的特征，从而可以控制曲线 $C(p)$ 演化，以从图像中分割出我们所需要的信息。比如图7.5(b)中，如果要检测灰色区域和白色区域的边界，可以设计属性函数，使其在图像的边界位置为零，其他位置则是较大的正值。比如，我们采用如下属性函数[1][9]

$$g[I(x,y)]=\frac{1}{1+|\nabla I(x,y)|^2}。\tag{7.1.11}$$

该函数在图像的边界有极小值，在其他位置都有较大的正值。通过上述属性函数，我们就可以控制闭合曲线 $C(p)$ 逐渐演化成位于边界的曲线 C_{Geo}，这样，就完成了图像的分割。

因此，应用几何曲线演化模型进行图像分割的基本思想就是耦合图像数据和曲线的变形速度(可以使用曲率变形或常量变形)，使得曲线演化停止在目标的边缘位置，演化过程通过水平集方法来实现。因此，对于几何变形模型的绝大多数研究都集中在如何设计演化的速度函数 $V(k)$。曲线演化方程(7.1.1)、(7.1.2)、(7.1.3)和(7.1.10)等的求解是和水平集方法结合起来的，与许多参数变形模型相比，用水平集方法解几何变形模型有不少优点。

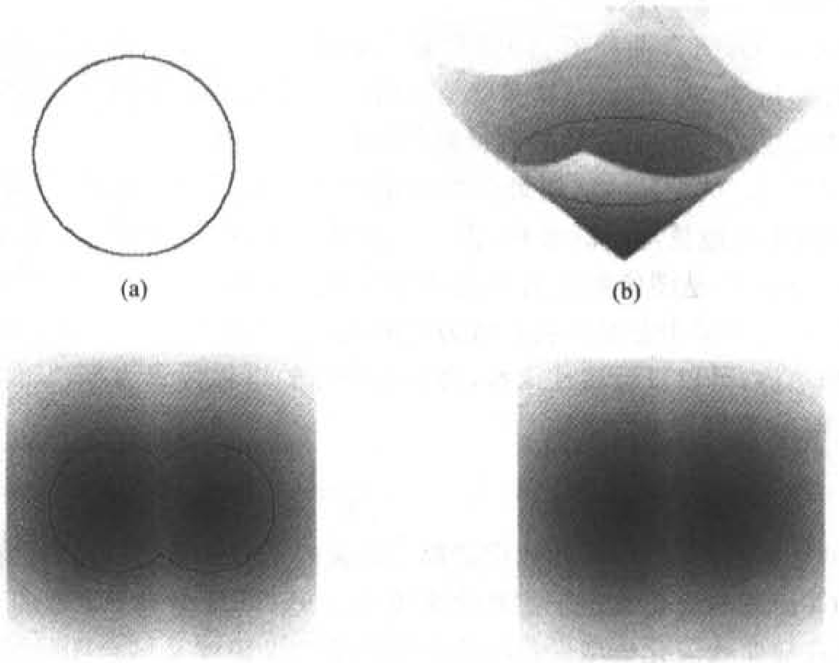
7.2 水平集方法概述

我们讨论了用于图像分割的几何活动轮廓线模型，该模型的最大特点是不依赖于活动轮廓线的参数化方式，因此可以很自然地处理曲线的拓扑结构的变化。不过，这些特征都是和水平集理论分不开的，可以说，几何活动轮廓线模型和水平集方法是相辅相成的。虽然我们也可以用参数化方法来表达几何活动轮廓线，然而用参数化方式计算曲线的曲率和法向矢量更加麻烦，并且仍然需要处理拓扑结构变化时重新参数化拓扑结构变换后的曲线，因此，需要更加自然地处理曲线的几何参数的计算方法，而水平集方法正好满足这些要求。

水平集方法(Level Set method)最初由 Osher 和 Sethian 提出[10][11][12]，用于解决遵循热力学方程下火苗外形的变化过程[13]，这是由于火苗外形的高动态性和拓扑结构变化的随意性，用参数化的曲线或曲面来描述火苗的这种变化显然是非常费力的。

平面闭合曲线被水平集方法隐含地表达为三维连续函数曲面 $\phi(x,y)$ 的一个具有相同函数值的同值曲线，通常是 $\{\phi=0\}$，称为零水平集，而 $\phi(x,y)$ 则称为水平集函数。图7.6说明了水平集函数表达闭合曲线的方式，其中图7.6(a)是一条平面闭合曲线，图7.6(b)是其水平集函数的隐含表达，图中的黑线就是隐含为水平集的平面曲线。

水平集方法处理平面曲线的演化问题不是试图去跟踪演化后的曲线位置，而是遵循一定的规律，在二维固定坐标系中不断更新水平集函数，从而达到演化隐含在水平集函数中的闭合曲线的目的。这种演化曲线方式的最大特点是：即使隐含在水平集函数中的闭合曲线发生了拓扑结构变化(合并或分裂)，水平集函数仍保持为一个有效的函数。如图7.6(c)、(d)所示，



(a) (b) (c) (d)

图 7.6 基于水平集方法的曲线演化示意图

(a) 平面曲线；(b) 水平集函数的隐含表达；

(c) 水平集函数的演化；(d) 曲线拓扑结构发生变化

其中图 7.6(c)是水平集函数更新后其零水平集曲线的形状，而图 7.6(d) 中的零水平集曲线已经分裂为两条。

下面首先简要说明水平集理论的数学表达及其和几何变形模型的关系，再介绍水平集方法的数值计算方法，随后讨论水平集方法实现中的一些问题，其中，讨论了如何加快水平集方法的计算速度，以及增强计算稳定性的问题。

7.3 水平集理论

设连续函数 $\phi(x,y,t):\Re^2\times\overline{\Re^-}\rightarrow\Re$ 是闭合曲线 $C(p,t):0\leqslant p\leqslant 1$ 在 t 时刻的隐含表达，即 t 时刻，$C(p,t)$对应于 $\phi(x,y,t)$的零水平集

$$\begin{cases}C(p,0)=\{(x,y)\mid\phi(x,y,0)=0\},\\C(p,t)=\{(x,y)\mid\phi(x,y,t)=0\}。\end{cases}\tag{7.3.1}$$

我们的目的就是要使水平集函数 $\phi(x,y,t)$在演化过程中，ϕ 的零水平集的平面闭合曲线

$$\phi[C(t),t]=0\tag{7.3.2}$$

始终满足曲线演化的偏微分方程(7.1.1)，即$\frac{\partial C}{\partial t}=V(k)\mathbf{N}$。对方程(7.3.2)求全微分，得

$$\nabla\phi\cdot\frac{\partial C}{\partial t}+\frac{\partial\phi}{\partial t}=0,\tag{7.3.3}$$

式中：$\nabla\phi$ 是 ϕ 的梯度。

设 s 是闭合曲线 C 的弧长参数，根据水平集函数的定义，ϕ 沿着曲线 C 的方向的变化量为零，即 $\phi_s=0=\phi_x x_s+\phi_y y_s=\langle\nabla\phi,C_s\rangle$。这样，$\nabla\phi$ 就垂直于闭合曲线 C 的切线 C_s，因此，$\nabla\phi$ 和 C

的法线同向。假设函数 ϕ 位于 C 内部的部分为负，外部部分为正，则水平集 C 的内向单位法向量就是

$$N = -\frac{\nabla\phi}{|\nabla\phi|}。\tag{7.3.4}$$

将式(7.1.1)和式(7.3.4)代入式(7.3.3)，可得

$$\phi_t = -\nabla\phi\cdot V(k)N = \nabla\phi\cdot V(k)\frac{\nabla\phi}{|\nabla\phi|} = V(k)\,|\nabla\phi|,$$

整理后，可得

$$\frac{\partial\phi}{\partial t} = V(k)\,|\nabla\phi|。\tag{7.3.5}$$

这样，给定几何活动轮廓线演化偏微分方程(7.1.1)以及任意水平集函数 $\phi_0(x,y):\phi_0(C_0)=0$，方程(7.3.5)可以保证水平集函数 $\phi(x,y,t)$ 随时间的演化满足 $\{\phi(C[(p,t),t]=0\}$ 的条件，即 ϕ 的零水平集始终是活动轮廓线 $C(p,t)$，上述方程称为曲线演化方程(7.1.1)的欧氏表达。方程(7.3.5)是一种 Hamilton-Jacobi 类型的偏微分方程[11]。

从上述方程(7.3.5)，可以很方便地计算出 $C(p,t)$ 的曲率[14]

$$k = \nabla\cdot\frac{\nabla\phi}{|\nabla\phi|} = \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2+\phi_y^2)^{3/2}}。\tag{7.3.6}$$

水平集方法实现活动轮廓线演化具有如下优点：

· 首先，只要速度函数 $V(k)$ 是平滑的，则水平集函数 $\phi(x,y,t)$ 始终保持为一个函数，然而，$\phi(x,y,t)$ 的水平集，即演化曲线 C 可以随着 ϕ 的演化很自然地改变其拓扑结构，可以分裂、合并、形成尖角等。这可以从图 7.6(c)和(d)看出。ϕ 所表达的闭合曲线 C 由图(c)中的一个闭合曲线变为图(d)中的两个闭合曲线，拓扑结构变了，水平集函数却仍然保持为一个函数。

· 因为 $\phi(x,y,t)$ 演化时始终保持为函数，因此很容易实现数值近似算法，可以利用离散网格结构的有限差分方法来很容易实现函数 $\phi(x,y,t)$ 的演化。

· 曲线 C 内在的几何特征，比如内向单位法矢和曲率等，可以利用式(7.3.4)和(7.3.6)直接由 $\phi(x,y,t)$ 计算出来。

· 水平集方法很容易扩展到高维情况，比如可以很方便地扩展到三维闭合曲面的演化，这对于三维图像的分割是非常有用的。

具体实现水平集方法时，还需要考虑如下几个问题：

(1) 水平集函数 ϕ 的选择。通常取由初始闭合曲线 C_0 生成的符号距离函数(Signed Distance Function)，记为 SDF。设 $\phi(\boldsymbol{x},t=0)$，$\boldsymbol{x}\in\Re^2$ 是 SDF，则

$$\phi(\boldsymbol{x},t=0) = \pm d(\boldsymbol{x}),\tag{7.3.7}$$

式中：$d(\boldsymbol{x})$ 是点 $\boldsymbol{x}$ 到初始曲线 $C(t=0)$ 的距离，其符号根据点 $\boldsymbol{x}$ 在闭合曲线 $C(t=0)$ 的内外部而定，如果 $\boldsymbol{x}$ 位于 $C(t=0)$ 的内部，则取正号(或负号)，反之取负号(或正号)，对于任意曲线，其符号距离函数的计算量比较大，因此，如何快速计算稳定的任意曲线的 SDF，对于提高水平集方法的效率和稳定性非常重要。后面我们还要具体讨论如何快速计算 SDF 的问题。

(2) 因为演化方程(7.3.5)仅仅从零水平集推导而来，一般速度函数 $V(k)$ 只在零水平集定义，其他水平集没有定义，不过，单位法矢和曲率在所有的水平集都有定义，因此，演化 ϕ 时，需要一种方法将 $V(k)$ 推广到所有的水平集，变为扩展的速度场 $V_{Ext}(k)$，比如，由图像的边缘信息所定义的边缘力，只是在零水平集才有定义，因此需要将零水平集的图像力扩展到 ϕ 的整

个定义域。然而，许多扩展速度场的方法常常导致水平集函数 ϕ 不再保持为 SDF，从而引起单位法矢和曲率计算的误差。因此，如何选择合适的方法扩展速度场，也是实现水平集方法时需要仔细考虑的问题。

(3) 在几何活动轮廓线的实际应用中，常常利用常量演化对活动轮廓线产生大尺度的变形作用。然而，常量演化可能使平滑的水平集函数产生尖角，一旦形成尖角，对 ϕ 进一步的演化就不确定了，因为在尖角位置，单位法矢的方向是不确定的。如何正确处理常量演化所导致的水平集函数奇异性的问题，也是实现水平集函数必须考虑的。

利用水平集表达方法，很容易将基于几何活动模型的演化方程表达为水平集的形式：
几何活动轮廓线方程

$$\frac{\partial\phi}{\partial t}=\alpha_{\text{img}}(k+V_0)\mid\nabla\phi\mid; \tag{7.3.8}$$

测地活动轮廓线方程

$$\frac{\partial\phi}{\partial t}=g(c_1+c_2k)\mid\nabla\phi\mid+\nabla g\cdot\nabla\phi; \tag{7.3.9}$$

测地活动区域方程

$$\frac{\partial\phi}{\partial t}=\alpha\log\left(\frac{p_B}{p_A}\right)\mid\nabla\phi\mid+(1-\alpha)[g(c_1+c_2k)\mid\nabla\phi\mid+\nabla g\cdot\nabla\phi]。 \tag{7.3.10}$$

7.4 水平集方法的数值计算

为了利用水平集方法实现曲线演化，有必要采用适当的数值计算方法。水平集方法的数值计算实现的关键是如何高效稳定地将定义于连续空间的偏微分方程(7.3.5)以离散形式表达出来。

由于水平集函数在演化过程中始终保持为一个函数，因此，我们可以用离散网格来表达水平集函数 $\phi(x,y,t)$。设离散网格的间隔为 h，且在 n 时刻，结点 ij 处的水平集函数为 ϕ_{ij}^n，则 $\phi_{ij}^n=\phi(ih,jh,n\Delta t)$。这里的 Δt 是时间步长，则方程(7.3.5)可以离散化为

$$\frac{\phi_{ij}^{n+1}-\phi_{ij}^n}{\Delta t}+V_{ij}^n\mid\Delta_{ij}\phi_{ij}^n\mid=0, \tag{7.4.1}$$

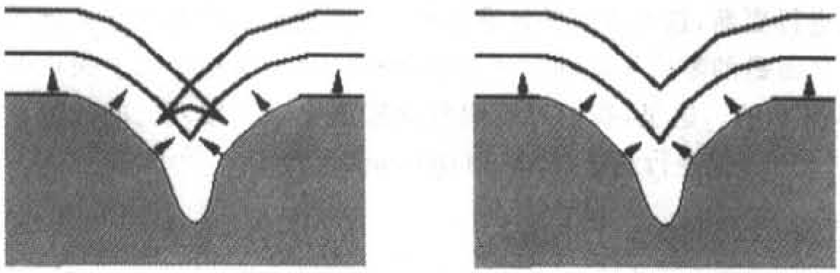
式中：V_{ij}^n 表示 n 时刻扩展速度函数 V_{Ext} 位于网格点 ij 处的值。要指出的是，上式中 $\phi(x,y,t)$ 的有限数值差分 $\Delta_{ij}\phi_{ij}^n$ 必须采取适当的形式，以避免常量演化中导致的水平集函数奇异性问题。论文[11][14]详细讨论了类似于方程(7.3.5)这种 Hamilton-Jacobi 类型的偏微分方程的数值解问题，正确处理此问题必须基于波前传播的"熵守恒"理论[13]。可以如此理解熵守恒的意义：假设演化的闭合轮廓线是正在纸上传播的火焰，则一旦某个位置被焰苗烧过，则该位置将一直保持烧过状态，不可能再次被燃烧，反映在曲线的传播上，就是曲线不会同时两次经过平面上一点，如图 7.7 所示的波前传播过程，图 7.7(a)是不适当的解，而图 7.7(b)是满足熵守恒的解。熵守恒原理应用在偏微分方程的求解上，就是偏微分方程的迟滞解方法。

为了介绍方程(7.3.5)的迟滞解方法，首先引入如下符号代表对 ϕ 的有限差分算子

$$D_{i,j}^{-x}=\phi_{i,j}-\phi_{i-1,j},D_{i,j}^{+x}=\phi_{i+1,j}-\phi_{i,j},D_{i,j}^{0x}=(\phi_{i+1,j}-\phi_{i-1,j})/2,$$

$$D_{i,j}^{-y}=\phi_{i,j}-\phi_{i,j-1},D_{i,j}^{+y}=\phi_{i,j+1}-\phi_{i,j},D_{i,j}^{0y}=(\phi_{i,j+1}-\phi_{i,j-1})/2。$$

方程(7.3.5)可以重写为



(a)　　(b)

图 7.7　波前传播的熵守恒原理

(a) 燕尾形解;(b) 熵守恒解

$$\phi_t + H(\phi_x, \phi_y) = 0, \tag{7.4.2}$$

这样的方程称为 Hamilton-Jacobi 方程，$H(\cdot)$称为哈米顿算子，论文[11]给出了上述方程的满足熵守恒的凸哈米顿数值解，本文在此罗列其一阶近似解。对于图像分割应用，方程(7.3.5)的速度函数常常可以写成

$$V = V_{\text{prop}} + V_{\text{curv}} + V_{\text{adv}}, \tag{7.4.3}$$

式中：$V_{\text{prop}} = V_0$ 是常量扩张速度(气压力)；$V_{\text{curv}} = -\varepsilon k$ 是曲率速度；$V_{\text{adv}} = \boldsymbol{U}(x,y,t)\cdot\boldsymbol{N}$ 是水平对流速度，则方程(7.3.5)的近似解为

$$\begin{aligned}\phi_{i,j}^{n+1} = \phi_{i,j}^{n} + \Delta t\{&-[\max(V_{oij},0)\,\boldsymbol{\nabla}^{+} + \min(V_{oij},0)\,\boldsymbol{\nabla}^{-}]\\ &-[\max(u_{ij}^{n},0)D_{ij}^{-x} + \min(u_{ij}^{n},0)D_{ij}^{+x}]\\ &-[\max(v_{ij}^{n},0)D_{ij}^{-y} + \min(v_{ij}^{n},0)D_{ij}^{+y}]\\ &+\varepsilon k_{i,j}[(D_{i,j}^{0y})^2 + (D_{i,j}^{0x})^2]^{1/2}\},\end{aligned} \tag{7.4.5}$$

式中：

$$\boldsymbol{\nabla}^{+} = [\max(D_{i,j}^{-x},0)^2 + \min(D_{i,j}^{+x},0)^2 + \max(D_{i,j}^{-y},0)^2 + \min(D_{i,j}^{+y},0)^2]^{1/2},$$

$$\boldsymbol{\nabla}^{-} = [\min(D_{i,j}^{-x},0)^2 + \max(D_{i,j}^{+x},0)^2 + \min(D_{i,j}^{-y},0)^2 + \max(D_{i,j}^{+y},0)^2]^{1/2}。$$

这种差分方法称为逆向有限差分法(Upwind Finite Difference)[11]。

通过式(7.4.5)，就可以利用迭代法来不断更新水平集函数，更新完成后，利用轮廓检测方法，提取更新后的水平集函数的零水平集，即可得更新后的闭合活动轮廓线。另外，闭合活动轮廓线的曲率可以直接采用中值差分方法来数值计算，即采用 D_{ij}^{0x} 和 D_{ij}^{0y} 来近似；而内向单位法矢可以按照下式计算：

$$\begin{aligned}n_{ij}^{*} = \frac{(\phi_x,\phi_y)}{\sqrt{\phi_x^2+\phi_y^2}} &= \frac{(D_{ij}^{+x}, D_{ij}^{+y})}{\sqrt{(D_{ij}^{+x})^2 + (D_{ij}^{+y})^2}} + \frac{(D_{ij}^{-x}, D_{ij}^{+y})}{\sqrt{(D_{ij}^{-x})^2 + (D_{ij}^{+y})^2}}\\ &+ \frac{(D_{ij}^{+x}, D_{ij}^{-y})}{\sqrt{(D_{ij}^{+x})^2 + (D_{ij}^{-y})^2}} + \frac{(D_{ij}^{-x}, D_{ij}^{-y})}{\sqrt{(D_{ij}^{-x})^2 + (D_{ij}^{-y})^2}},\end{aligned} \tag{7.4.6}$$

然后，

$$\boldsymbol{N}_{ij} = \frac{n_{ij}^{*}}{|n_{ij}^{*}|}。 \tag{7.4.7}$$

7.5　水平集函数演化的快速算法

直接根据上一节的数值方法计算水平集函数的演化，需要对整个图像定义域中的所有点

的水平集函数值进行更新，这种方式的计算量很大，特别是在图像比较大的情况下。其实，我们仅仅关心水平集函数的零水平集的演化过程，因此，如果对整个水平集函数都更新，显然有许多计算都是不需要的。这里，我们讨论两种快速演化水平集方法的数值计算方法：窄带法(Narrow Band)[15][16]和快速行进法(Fast Marching)[12][17]。

7.5.1 窄带法(Narrow Band)

窄带法最初由 D. Chop 提出[15]，Adalsteinsson 和 Sethian[16]对该方法给出了详细的实现。窄带法的基本思想是只演化水平集函数位于零水平集周围很窄的一个带状区域的值。如图7.8所示，阴影条带中的闭合曲线是当前零水平集，而外部的曲线则是窄带的外边界点，内部曲线是内部边界点，更新水平集函数的时候，只对带状阴影内部的激活点更新。

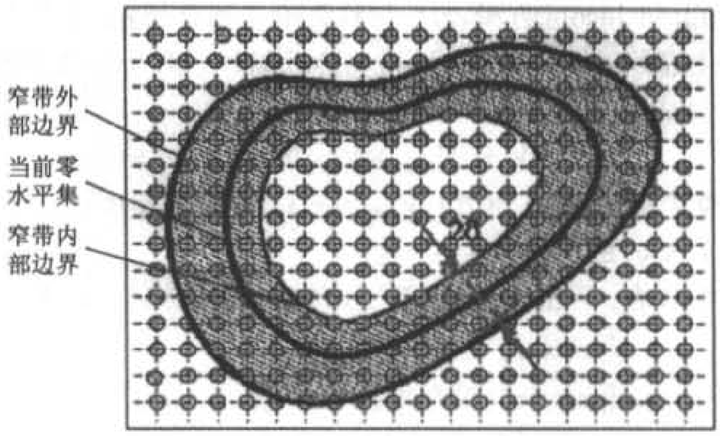


图 7.8 窄带法示意图

由于窄带宽度一般较窄，窄带内需要更新的网格点(激活点)不多，因此，更新水平集函数的计算量大大减小。不过，窄带法存在的问题是由于零水平集的移动，经过几次迭代后，可能超出窄带的范围，因此，需要经常更新窄带状水平集函数。更新的项目包括：更新窄带的内外部边界点，以保持窄带的有效范围；重新初始化水平集函数，保持水平集函数为符号距离函数。更新的方式和窄带的宽度有关，如果窄带宽度很窄，则一次迭代零水平集曲线就有可能超出窄带范围，需要每次更新，这将增加计算量；如果窄带宽度很宽，则需要更新的窄带内的网格点比较多，计算量也比较大；比较合理的窄带宽度是 $2\delta=12\sim16$。

下面给出窄带法更新水平集函数的基本流程：

(1) 构造窄带。

① 由初始闭合曲线 C_0，构造窄带内的符号距离函数，将窄带内的所有点标记为激活点(Active)；

② 标记窄带的边界点(Boundary)；

③ 将远离窄带的点标记为远点(Faraway)，并对 C_0 以外的远点赋予较大的正值，C_0 内部的远点赋予较大的负值。

(2) 迭代。

① 据方程(7.4.5)，更新激活点的 ϕ；

② 检测更新后 ϕ 的零水平集曲线 C^n，检查是否到达边界点，如果到达，则根据当前 C^n，按照步骤(1)，重新构造窄带和 SDF，然后继续迭代；

③ 收敛检查。

7.5.2 快速行进法(Fast Marching)

这里，我们考虑曲线演化的特殊情况，即方程(7.1.1)中的速度 $V(\boldsymbol{x})$ 总是保持符号不变，即总是正值，或总为负值，这时方程(7.1.1)称为静态 Hamilton-Jacobi 方程，因为时间变量可以消除。Sethian 提出了求解静态 Hamilton-Jacobi 方程边界值问题的快速行进法(Fast Marching)[14][17]。如果以 $T(\boldsymbol{x})$ 表示零水平集通过点 $\boldsymbol{x}$ 的时间边界，则 $T(\boldsymbol{x})$ 满足方程

$$|\nabla T| V = 1。\tag{7.5.1}$$

利用逆向差分法(Upwind)，方程(7.5.1)的稳定解可由如下方程得到：

$$\left[\max(D_{i,j}^{-x}T,0)^2 + \min(D_{i,j}^{+x}T,0)^2 + \max(D_{i,j}^{-y}T,0)^2 + \min(D_{i,j}^{+y}T,0)^2\right]^{\frac{1}{2}} = 1/V_{i,j},\tag{7.5.2}$$

式中：D^- 和 D^+ 分别是后向与前向差分，例如 $D_{i,j}^{-x}T = \frac{T_{i,j} - T_{i-1,j}}{h}$，$D_{i,j}^{+y}T = \frac{T_{i,j+1} - T_{i,j}}{h}$；$h$ 是网格大小。

由方程(7.5.2)可知，边界传播方向是从 T 比较小的点流向 T 比较大的点，根据这样的特点，Sethian 提出了快速行进法来迅速传播边界值 T。基本思想是在传播边界外围构造一个激活窄带，窄带内的点的到达时间未定，当前传播边界利用逆向格式(Upwind Scheme)[17]将当前边界向外传播，就像水波扩散一样，凡是扩散到的点，就冻结其波前到达时间，然后再根据当前的波前构造新的激活带，如此循环，就可以得到整个平面上每点的到达时间。下面我们具体介绍快速行进法的计算步骤。

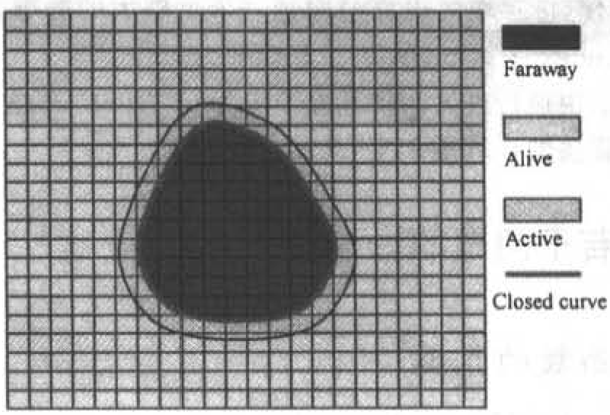


图 7.9　快速行进法初始化示意图

根据图像中每点距离闭合曲线 C_0 的远近，将图像中所有的点分成三类(如图 7.9 所示)：

- Alive 点：已经被波前传播过的点，到达时间已知。
- Active 点：当前波前即将经过的点，到达时间未知。
- Faraway 点：远离波前的点，到达时间未知。

根据上述定义，快速行进法的基本步骤如下：

(1) 初始化。

① 将 C_0 所在网格点标记为"Alive"，并且时间 $T=0$；

② 考察每个"Alive"点的四邻点，如果有未被标记为"Alive"的，则以"Active"标记之，并

赋予到达时间 $T_{ij}=1/V_{ij}$，并将所有标记为 Active 的点放入一个排序堆中，排序堆按照每点的 T 从小到大的顺序排序；

③ 剩余的既不是“Active”也不是“Alive”标记的其他点，标记为“Faraway”，到达时间标记为 $T_{ij}=\infty$。

(2) 前向行进。

① 输出 T 最小的 Active 点：从排序堆输出窄带区中 T 最小的点 $P_{\min}(i_{\min},j_{\min})$，标记为 Alive，并从排序堆删除；

② 更新邻点：选择点 $P_{\min}$ 的四连通邻点 $(i_{\min}-1,j_{\min})$，$(i_{\min},j_{\min}-1)$，$(i_{\min}+1,j_{\min})$，$(i_{\min},j_{\min}+1)$ 中标记不是 Alive 的点，计算各点的到达时间 T；计算时，要在待计算点的四邻接点中，选择标记为 Alive 的点的 T 代入方程(7.5.2)。计算完毕，如果邻点的当前标记为 Faraway，则将其标记修改为 Active，插入排序堆中；如果邻点的标记是 Active，则利用新的 T 值更新该点的 T，并重新调整该点在排序堆中的位置，保持排序堆的顺序；

③ 收敛检查：如果 $T_{P_{\min}}<T_{\text{threshold}}$，或排序堆空，则行进结束，否则重复行进。

上面的计算中，快速排序堆的使用是提高算法效率的关键。排序堆的结构要注意以下几点：

- 排序堆内部的逻辑结构是对称二叉树，内部存放 Active 点坐标及其 T，每个根节点的 T 总比其两个子节点的小，即 $T_f\leqslant\min(T_{c1},T_{c2})$，子节点则按照进入堆的先后顺序放置。
- 排序堆中的元素和 $T(\boldsymbol{x})$ 二维平面中的点通过双向指针联系，即不仅排序堆中每个元素存储 $T(\boldsymbol{x})$ 的 Active 点坐标，而且 $T(\boldsymbol{x})$ 的 Active 点有一个反向指针，指向该点在排序堆中的位置，这样可以使得查询和更新的计算复杂度仅为 $O(1)$。
- 为了算法实现方便，排序堆的内存结构是一个足够大的静态数组，各个元素位置之间满足如下关系 $\text{index}_{c1}=\text{index}_f\cdot 2$，$\text{index}_{c2}=\text{index}_{c1}+1$。

论文[17]详细分析了快速行进法的计算复杂度为 $O(N\log N)$，这里 N 为图像的点数。这比水平集的直接数值计算法的计算复杂度 $O(N^2)$ 小多了。

7.6 数值计算中若干问题的讨论

7.6.1 符号距离函数的构造

符号距离函数的构造分两部分：首先构造符号表 $S_{C_0}(p)$，根据闭合初始轮廓线 C_0，区分其内外部，并以正号(或负号)标记 C_0 内部，即 $S[\text{inside}(C_0)]=+1$，以负号(或正号)标记 C_0 外部，即 $S[\text{outside}(C_0)]=-1$；其次，计算图像定义域中每一个网格点 p 到 C_0 的最短距离 $d_{C_0}(p)$，则

$$\phi_0(p)=S_{C_0}(p)d_{C_0}p。\tag{7.6.1}$$

为了增强计算符号距离函数的稳定性并提高计算速度，本节提出一种采用快速行进法构造符号表 S 的方法；并改进一种基于源点映射快速构造距离函数的方法，只需要对所有网格点四次扫描和简单的运算，就可以完成距离函数的构造。

1. 初始符号表的构造

区分初始闭合轮廓线 C_0 的内外部是构造符号距离函数的第一步。常用的方法是选取一

些规则的闭合曲线，如矩形或圆等[14][22]。由于已知这些规则闭合曲线的几何参数，因此很容易知道其内外部。然而，对于任意形状的初始闭合曲线，要区别其内外部就比较繁琐，逐行扫描的方式需要复杂的判断，而且容易出现漏洞，而一旦出现漏洞，将影响整个符号距离函数构造的正确性。

本节提出一种利用快速行进法[17]标记 C_0 内外部的方法，该方法需要若干个位于 C_0 内部的种子点 P_{seed} 作为行进的源点，并将 C_0 所经过的网格点设为停止点，然后从种子点开始行进，行进访问过的点都标记为某种符号，直到访遍 C_0 内部所有网格点为止。行进时，采用行进方程

$$[\max(D_{i,j}^{-x}T,0)^2+\min(D_{i,j}^{+x}T,0)^2+\max(D_{i,j}^{-y}T,0)^2+\min(D_{i,j}^{+y}T,0)^2]^{\frac{1}{2}}=1, \tag{7.6.2}$$

式中：$T_{i,j}$ 表示从种子点 P_{seed} 到达考察点 $P_{i,j}$ 的时间。上述方程其实就是方程(7.5.2)在 $V=1$ 时的特殊情况。这种方法可以同时构造若干个任意复杂度的闭合曲线的符号表。这样，可以利用这种方法，采用多个闭合曲线初始化水平集函数，这种初始化方式将大大加快水平集方法的计算速度。具体算法和前面的快速行进法基本相同，只是初始化和结果处理有些差异，简要步骤如下：

(1) 初始化。

① 设 C_0 上的所有点 $T=+\infty$，并标记为 Alive；

② 设种子点 P_{seed} 的 $T=0$，也标记为 Alive；种子点周围的四个四邻接点的 $T=1$，标记为 Active，并按 T 从小到大顺序放入二叉排序堆 $Heap(T)$；

③ 其他所有的网格点到达时间 $T=-\infty$，标记为 Faraway。

(2) 前向行进循环。

① 从 $Heap(T)$ 中输出 T 最小的点 $P_{\min_T}$，标记为 Alive，并从 $Heap(T)$ 中删除；

② 由方程(7.6.2)，计算 $P_{\min_T}$ 的四个四邻接点中标记为 Active 或 Faraway 点的 T，如果某个四邻点的标记是 Active，则更新它在 $Heap(T)$ 中的 T 值；如果标记是 Faraway，则将其加入 $Heap(T)$；

③ 如此循环，直到 $Heap(T)$ 中没有元素为止。

(3) 输出。将 $T\geqslant 0$ 的所有点以 $+1$ 标记，其他的点以 -1 标记。

这样，我们就得到了 C_0 内部为正号、外部为负号的符号表。根据此符号表，就可以方便地构造 SDF。

2. 距离函数的快速构造

根据 C_0 计算每点 p 的距离函数时，由于计算 p 到 C_0 的距离的复杂度为 $O(M)$，这里 M 是 C_0 在网格上的点数，则总的计算复杂度为 $O(MN)$，N 是网格点数。计算量很大，特别是初始轮廓线比较长的时候。而在某些情况下，我们需要经常对距离函数进行更新，比如窄带法中需要经常重构符号距离函数；某些速度函数会使得符号距离函数逐渐变形，这时也需要重新构造距离函数。因此，如何快速构造距离函数，对于提高水平集方法的计算速度非常重要。

Sethian 等提出了利用快速步进法构造距离函数的方法[14][18]。我们知道，符号距离函数 ϕ 所满足的偏微分方程为

$$|\nabla\phi|=1。\tag{7.6.3}$$

上述方程就是快速步进法的偏微分方程(7.5.1)在 $V=1$ 时的特殊情况。因此，可以利用快速

步进法分两次来直接构造符号距离函数：首先利用快速步进法，构造 C_0 的外部网格点的距离函数(设 C_0 所在网格点的到达时间为 $T(C_0)=0$)；对外部网格点标记后，再对 C_0 的内部网格点快速步进，这样，分两次就可以得到所有网格的符号距离函数值，这种方法的计算量是 $O(N\log N)$。不过，这种方法所计算出来的 ϕ 值不太准确，比如设某点 (i,j) 的 $\phi_{ij}=0$，从该点用快速行进法向其四周传播波前，则只有其四邻点的距离函数是正确的，即 $\phi=1$，而其八邻接点的 $\phi=2$，真实值却是$\sqrt{2}$。

另外一种方法就是利用 Chamfer 距离计算方法[19][20]。这种方法以模板双向扫描来计算距离函数。首先设 C_0 上的网格点的 $\phi_0=0$，其他点的 $\phi_0=\infty$，然后采用图 7.10 所示的两个模板来计算距离函数。

$\sqrt{2}$	1	$\sqrt{2}$
1	0	

前向模板

	0	1
$\sqrt{2}$	1	$\sqrt{2}$

后向模板

图 7.10　Chamfer 距离函数模板

前向模板从距离函数网格左上角向右下角的方向扫描所有网格，后向模板则从右下角向左上角的方向扫描，当前网格的距离函数值按下式更新：

$$\phi_0(x,y)=\min_{-1\leqslant i,j\leqslant 1} m(i,j)+u(x+i,y+j),\tag{7.6.4}$$

式中：$m(i,j)$是模板的值。这种方法速度虽然快，但是和快速行进法一样，计算误差仍然很大。

下面，我们首先介绍一下 Tsai 提出的源点扫描法，然后对该方法进行改进，简化计算。实验表明，改进的源点扫描法不仅计算速度快(计算复杂度为 $O(N)$)，而且计算精确、简单。

Tsai[21]提出的源点扫描法确认与每个网格点 P 同处一条特性线(轮廓线 C 的法线)上的轮廓线点，即 P 点基于 C 的源点 P_S，源点是 C 上距离待考察网格点最近的点，从而可计算出 P 点到 P_S 点的距离，就是 P 点的距离函数值，这样就可以快速构造距离函数。为了叙述源点映射的原理，先定义如下概念：

定义 1　设 Ω 是 $\Re^2$ 上的连通子集，点 $P\in\Omega$，并且点集 $\Gamma\subset\Omega$，定义点 P 基于 Γ 的源点为 $\mathrm{Source}(P)=\{Q\in\Gamma:\mathrm{dist}(P,\Gamma)=|P-Q|\}$，并以 P^* 代表，又称为 P 的 Voronoi 源。

从定义可以看出，P 点在 Γ 上的源点 P^* 就是 Γ 上距离 P 点最近的点。

定义 2　(二维扫描方向)定义下列规则排列二维网格$(1:Nx-1,1:Ny-1)$上的扫描迭代方向：

$(x+,y+)$扫描：从左到右，从上到下；$(x-,y+)$扫描：从右到左，从上到下；

$(x+,y-)$扫描：从左到右，从下到上；$(x-,y-)$扫描：从右到左，从下到上。

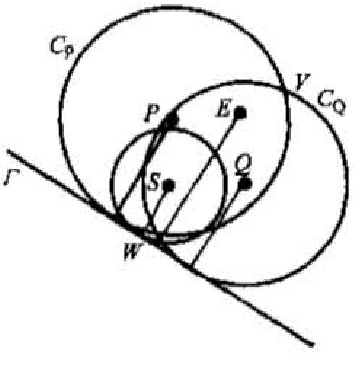


图 7.11　源点映射一致性检查

Tsai 的方法基于这样的先验知识，即假设点 E(如图 7.11 所示)是当前考察点，要计算距离函数 $u(E)$，其源点 E^* 未知。设 P、Q 是两个 X 和 Y 方向的四邻接点，这两点到 Γ 的距离已知且是各自方向距离 Γ 最小的点，则必有 $dist(E,E^*)>\max(u(P),u(Q))$。这其实就是快速行进法中的逆向条件(Upwind condition)。Tsai 的方法，就是在每次扫描的时候，保持上述条件，并计算以 P 和 Q 为中心，$u(P)$和 $u(Q)$为半径的两个圆 C_P 和 C_Q 的交点 V 和 W，则 $u(E)$ 可以近似为 $u(E)=\max\{|E-V|,|E-W|\}$。对所有网格进行四个方向的扫描后，即

可得到距离函数 $u(x,y)$。不过，因为需要求取圆的交点等，这种方法的计算仍然比较复杂，并且计算仍然存在一定的误差。

其实，可以首先计算出 Γ 及其最近点之间的距离，并保存每点的源点，然后遵循上述逆向条件对所有网格点扫描，比较每点与其四邻点的源点的距离，并记录距离最短的四邻点的源点，作为当前点的源点保存起来，这样就可以大大降低寻找源点的计算量，提高计算速度，并且，由于保存了每点在 Γ 上的源点映射，还可以用于水平集方法中将零水平集的速度函数扩展到所有水平集。

本文提出的源点扫描改进方法构造基于点集 Γ(即 C_0 的网格点)的距离函数步骤如下：

(1) 初始化。计算距离点集 Γ 最近的网格点 $P_{closed}=\{p\in\Omega: 0\leqslant dist(p,\Gamma)\leqslant 1\}$ 到 Γ 的准确距离 d 以及相应的源点 P^*，并保存最近点的 d 到距离函数数组 u，源点 P^* 到映射数组 $SMap$，将所有最近点标记为 Visited；其他点的 $d=+\infty$，标记为 Unvisited。

(2) 分别以四个扫描方向 $\{(x+,y+),(x-,y+),(x+,y-),(x-,y-)\}$ 扫描网格 Ω。每次扫描中，更新每个网格点 E(不包括标记为 Visited 的点)的距离函数值 u_E 及其最近点映射值 V_E，更新方法如下：

① 计算点 E 与每个四邻接点 P_i 源点 P_i^* 的距离 $u_i^{tmp}=\| E-P_i^* \|$；

② 如果 $u_i^{tmp}<\min_i u(P_i)$，则设 $u_i^{tmp}=+\infty$，保证距离函数的单调性；

③ 设置 $u(E)=\min_i u_i^{tmp}=u_k^{tmp}$ 并且 $SMap(E)=P_k^*$。

整个计算只需要扫描四次，就可以计算出 Ω 的所有网格点的距离函数值，因此计算复杂度为 $O(N)$，速度非常快，而且由于所计算的距离值是每点及其源点之间的距离，因此精度较高，这可以从后面的实验看出。

按照上述方法根据 C_0 构造出符号表和距离函数后，利用式(7.6.1)，就可以得到 C_0 的符号距离函数。

7.6.2　速度场的扩展

前面已经提到，利用水平集方法解决曲线演化问题时，需要将零水平集的速度函数扩展到整个网格域。最简单的方法就是计算每个网格 P 到 C_0 上距离最短的点 $P^*\in C_0$，然后以该点的速度作为 P 点的速度 $V_P=V_{P^*}$。Malladi 的论文[1]中详细论述了这种方法。不过，按照这种方法计算所有网格点的速度，由于需要寻找和 C_0 上距离最短的点，计算量仍然较大。另外一种方法是基于水平集函数为符号距离函数，因此到任意网格点距离最短的 C_0 上的点就是 $p^*=p-|u(p)|\nabla u(p)$，这样就得到 p^* 点，然而由于距离函数可能存在误差，由这种方法所求的 p^* 可能存在位置误差，特别是 p 点距离 C_0 比较远的情况下，误差更大。

论文[18]提出了一种利用快速行进法同时构造符号距离函数和扩展速度场的方法，这种方法基于如下偏微分方程：

$$\nabla\phi\cdot\nabla V_{Ext}=0。\tag{7.6.5}$$

研究表明，如果扩展速度场 V_{Ext} 以及水平集函数 ϕ 之间满足上述关系，则 ϕ 将一直保持为符号距离函数。上述偏微分方程的意义很明显，因为 ϕ 满足 $|\nabla\phi|=1$，即 ϕ 沿着法线方向变化，而切线方向 ϕ 的变化率为零；根据方程(7.6.5)，扩展速度场 V_{Ext} 的变化刚好和 ϕ 的相反，即 V_{Ext} 沿着法线方向不变，只在切线方向有变化，这正是对扩展速度场的要求。因此，可以在利用快速行进法构造 ϕ 的同时，利用方程(7.6.5)，同时构造 V_{Ext}。这种方法的优点是可以一次完成 ϕ

和 V_{Ext} 的构造，计算复杂度和快速行进法相同，即 $O(N\log N)$，N 是需要进行速度场扩展的网格点数目。不过，这种方法的计算量仍然不小，况且快速行进法所构造的符号距离函数本来就存在误差。

本文提出利用源点扫描构造扩展速度场的方法，由上一节的源点扫描法可知，源点扫描法在构造距离函数的同时，可以得到每个网格点 P 在 C_0 上的对应点——源点。源点就是和 P 同在一条特征线（C_0 的法线）上的点，因此，在利用源点扫描法构造完距离函数后，利用 $SMap$ 数组，就可以计算每一网格点的速度，简要步骤如下：

① 源点扫描法初始化时，同时计算零水平集上网格点的速度；

② 源点扫描法获得源点映射函射 $SMap(p)$；

③ 根据 $V(p)-V(SMap(p))$，即可得扩展速度场 V_{Ext}。

7.6.3 迭代时间间隔的选取

基于方程(7.4.5)利用迭代法进行水平集函数演化时，必须审慎选择时间间隔 Δt，以保证偏微分方程数值计算的稳定性。Sethian 在其论文[18]中论述了时间间隔的选择必须满足如下 CFL 条件：

$$\Delta t \max V \leqslant \Delta x, \tag{7.6.6}$$

式中：Δx 表示网格间隔。上式的含义是每次迭代时，零水平集的移动范围不能超过一个网格宽度。实际计算时可以扫描速度场，按照上式动态设定迭代时间间隔。

7.6.4 符号距离函数的构造方法性能比较实验

下面通过对比几种符号距离函数求取方法的计算效率，来证明本章所提出的改进源点扫描法计算距离函数的高效率。实验首先采用本章所提出的符号标记法得到任意形状闭合曲线的内外部标记图，然后分别利用直接求距离函数方法（即计算网格中每一点到闭合曲线的最短距离，作为该点的距离函数值，再加上符号图中的符号，作为该点的 SDF 值）、快速行进法以及本章方法计算不同大小网格、不同点数的闭合曲线的距离函数，并记录计算的时间以及与直接法所求 SDF 值的方差。这里，取网格的大小分别为 128×128、256×256 以及 512×512。

图 7.12(a)是任意形状的闭合曲线，其中的点为种子点；图 7.12(b)是利用本文方法所计算的符号图，内部为正值，外部为负值；图 7.12(c)为对应的符号距离函数（内正外负型）。多次实验表明，本文提出的符号表生成方法是非常鲁棒的，而且速度也很快。

表 7.1 是直接法、快速行进法和本章所改进源点扫描法构造 SDF 的计算时间对比，表中的方差表示快速行进法和改进源点扫描法所得 SDF 与直接法的 SDF 之间的平均方差值。平均方差按如下方式计算：设直接法所得 SDF 为 ϕ_D，其他方法所得 SDF 为 ϕ_O，SDF 的网格点数为 N，则 $E=\frac{1}{N}\sum_{i}^{N}|\phi_D(p_i)-\phi_O(p_i)|^2$。从表中可以看出，改进源点扫描法的计算速度最快，而且计算精度和直接法相差非常小，精度很高。另外，本章方法和轮廓线的点数没有直接关系，这样就不会出现类似直接法中当轮廓线点数较多时计算变慢的现象。



(a)　(b)　(c)

图 7.12　本文方法所产生的基于任意形状闭合曲线的符号距离函数

(a) 任意形状闭合曲线；(b) 符号图；(c) 符号距离函数

表 7.1　三种 *SDF* 构造方法比较

方　法	网格大小	轮廓线点数	计算时间/s	平均方差
直接法	128×128	208	0.701	—
		1390	4.617	—
	256×256	361	4.807	—
		1868	24.705	—
	512×512	408	21.691	—
		2525	134.994	—
快速行进法	128×128	208	1.032	46.811
		1390	2.494	5.430
	256×256	361	6.699	171.998
		1868	12.698	43.408
	512×512	408	45.695	484.205
		2525	86.765	109.899
改进源点扫描法	128×128	208	0.11	0.005
		1390	0.100	0.015
	256×256	361	0.440	0.005
		1868	0.450	0.008
	512×512	408	1.823	0.002
		2525	1.893	0.004

7.7　Mumford-Shah 模型

如前所述，基于活动轮廓线模型并以水平集方法表达的图像分割偏微分方程为

$$\begin{cases}\dfrac{\partial \phi}{\partial t}=g[\,|\nabla I(C)|\,](k+v)\,|\nabla \phi|, \\ \phi(\boldsymbol{x},0)=\phi_0(\boldsymbol{x})。\end{cases} \tag{7.7.1}$$

这个模型依赖于基于图像梯度$|\nabla I|$的边缘函数 $g(\cdot)$来停止活动轮廓线的演化过程，即当轮廓线靠近图像的边界，$|\nabla I(C)|$增大，导致 $g(\cdot)\to 0$，则曲线的速度降为零，停止运动。这样，曲线就停靠在图像的边缘位置。然而，由于图像中的边缘并非都是理想的阶梯边缘，如果轮廓线位于边缘强度较弱的平滑边缘，则可能越过边缘，出现“冒顶”现象，且不会返回到正确位置。虽然 Caselles[9] 等引入另外一些额外约束项，能一定程度克服轮廓线“冒顶”的现象，然而却不能根本上解决问题，关键原因就是这些方法仅依靠位于闭合轮廓线 C 下的图像局部

信息来控制其运动，只见树叶不见森林，难以获得同质区域的全局分割效果。另外，如果图像 I 的噪声很大，则需要方差很大的高斯滤波器，这样常常模糊图像中边缘，即使活动轮廓线停止，可能也会出现定位不精确的现象。

Chan 和 Vese 提出了一种基于 Mumford-Shah 模型的活动轮廓线图像分割方法，这种方法的停止函数不再依赖于图像的局部梯度，而是基于 Mumford-Shah 分割模型。这样，该方法同时适用于有梯度和没有梯度的轮廓的检测，比如那些边缘很平滑或边缘不连续的图像的分割。另外，该方法的模型定义于所有水平集，因此，可以检测出带有空洞目标的内部区域。

以下介绍简化的 Mumford-Shah 图像分割模型。

设图像 $I(x,y)$ 的定义域为 Ω，且当前考察的图像边缘 C 对图像 $I(x,y)$ 将图像划分为若干近似同值区域，得分割图像 $I_o(x,y)$，则 Mumford-Shah 图像分割模型就是寻找真正的图像边界 C_o。将图像 $I(x,y)$ 划分为若干同质区域，并且所得分割图像 $I_o^{MS}(x,y)$ 和 $I(x,y)$ 的误差比所有分割图像和原图像的误差都要小，即最小化如下能量方程：

$$(C_o, I_o^{MS}) = \arg\min F^{MS}(I_o, C) \tag{7.7.2}$$

$$F^{MS}(I_o,C) = \mu Length(C) + \lambda\int_{\Omega} |I - I_o|^2 \mathrm{d}x\mathrm{d}y + \int_{\Omega/C} |\nabla I_o|^2 \mathrm{d}x\mathrm{d}y$$

上式的意义很明显，当 $F^{MS}(I_o,C)$ 最小时，所得边界 C_o 将图像划分为若干个平滑区域，并且保留了尖锐的边界 C_o。

Chan 和 Vese 提出了一种 Mumford-Shah 的简化分割模型，即假设图像中每个同质区域的灰度是常数，也即对于区域 $R_i \subset \Omega/C$，有 $I(R_i)=c_i$，c_i 是常数，则最小化能量函数 F^{MS} 的目的，就是寻找最优分割 C_o，使得分割图像。

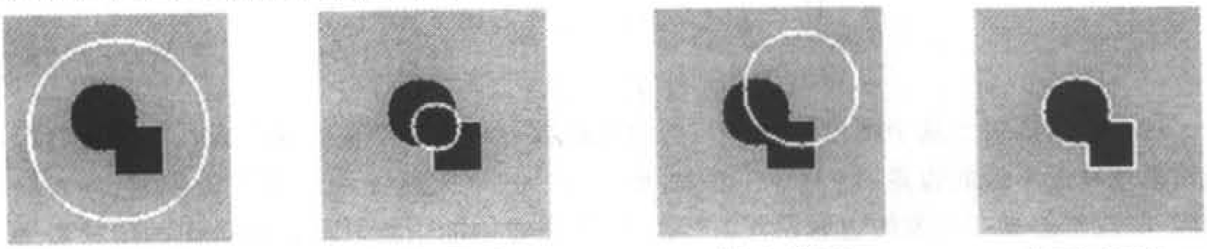
$$I(R_i) \approx \begin{cases} C_o, & \text{如果 } R_i \in \text{inside}C; \\ C_b, & \text{如果 } R_i \in \text{outside}C. \end{cases} \tag{7.7.3}$$

和原图像 I 之间的差异最小。

下面首先从一个简单的图形分割来分析 Chan-Vese 提出的分割模型的意义。设原图像 $I(x,y)$ 被活动轮廓线 C 划分为目标 ω_o 和背景 ω_b 两个区域。各个区域的平均灰度为 c_o 和 c_b，Chan 和 Vese 所提出的简化的基于 Mumford-Shah 图像分割模型的拟合能量函数如下：

$$F(C) = F_o(C) + F_b(C) = \int_{\text{inside}(C)} |I - c_o|^2 \mathrm{d}x\mathrm{d}y + \int_{\text{outside}(C)} |I - c_b|^2 \mathrm{d}x\mathrm{d}y$$

上式中的 C 是任意闭合活动轮廓线。可以看出，当闭合活动轮廓线 C 没有位于两个同质区域的边界 C_o 时，$F(C)$ 不能达到最小值，如图 7.13 中前三种情况所示；只有轮廓线位于两个同质区域边界时，$F(C)$ 才能达到最小值。



$F_o>0, F_b\approx0$ $F(C)>0$　　$F_o\approx0, F_b>0$ $F(C)>0$　　$F_o>0, F_b>0$ $F(C)>0$　　$F_o\approx0, F_b\approx0$ $F(C)\approx0$

图 7.13 C-V 分割模型示意图

据此，Chan 和 Vese 提出了如下的图像分割能量函数：

$$F(C,c_o,c_b)=\mu L(C)+vS_o(C)+\lambda_o\int_{\text{inside}(C)}|I-c_o|^2\mathrm{d}x\mathrm{d}y+\lambda_b\int_{\text{outside}(C)}|I-c_b|^2\mathrm{d}x\mathrm{d}y,\quad(7.7.4)$$

式中：$L(C)$是闭合轮廓线 C 的长度；$S_o(C)$是 C 的内部区域面积；$\mu,v\geqslant0,\lambda_o,\lambda_b>0$，是各个能量项权重系数；$F$ 的头两项是平滑项。最终分割轮廓线 C 的位置以及未知数 c_o,c_b 经最优化上式得到，即

$$\{C^o,c_o^o,c_b^o\}=\inf_{C,c_o,c_b}F(C,c_o,c_b)。\quad(7.7.5)$$

由于上面的模型利用了图像全图信息，因此通过最优化能量函数式(7.7.4)可以得到全局最优的图像分割结果。

7.8 求解分割模型的 C-V 方法及其改进

7.8.1 求解 M-S 图像分割模型的 C-V 方法

设 ϕ_0 是根据初始轮廓线 C_0 构造的符号距离函数，即$\{C_0\mid\phi_0(x,y)=0\}$，并设 ϕ 为内正外负型的 SDF，即 $\phi[\text{inside}(C)]>0,\phi[\text{outside}(C)]<0$。可以证明，以水平集函数表达的轮廓线 C 的长度和轮廓线内部的面积[14]分别为：

$$L(C)=\int_\Omega|\nabla H|\mathrm{d}x\mathrm{d}y=\int_\Omega\delta(\phi)|\nabla\phi|\mathrm{d}x\mathrm{d}y,$$

$$S[\text{inside}(C)]=\int_\Omega H(\phi)\mathrm{d}x\mathrm{d}y,\quad(7.8.1)$$

式中：Ω 是水平集函数的定义域；$H(z)$是 Heaviside 函数，即

$$H(z)=\begin{cases}1 & \text{if } z>0\\ 0 & \text{if } z<0\end{cases},$$

而 $\delta(x)$则是 Dirac 函数。根据式(7.8.1)，可以将能量函数式(7.7.4)以水平集函数 ϕ 表达：

$$F(\phi,c_o,c_b)=\mu\int_\Omega\delta(\phi)|\nabla\phi|\mathrm{d}x\mathrm{d}y+v\int_\Omega H(\phi)\mathrm{d}x\mathrm{d}y+\lambda_o\int_\Omega|I-c_o|^2H(\phi)\mathrm{d}x\mathrm{d}y+\lambda_b\int_\Omega|I-c_b|^2[1-H(\phi)]\mathrm{d}x\mathrm{d}y。\quad(7.8.2)$$

Chan 和 Vese 在其论文中以欧拉-拉格朗日方法推导出了求解式(7.8.2)，并以水平集函数 ϕ 表达的偏微分方程如下：

$$\begin{cases}c_o(\phi)=\dfrac{\int_\Omega I(x,y)H_\varepsilon(\phi)\mathrm{d}x\mathrm{d}y}{\int_\Omega H_\varepsilon(\phi)\mathrm{d}x\mathrm{d}y},c_b(\phi)=\dfrac{\int_\Omega I(x,y)[1-H_\varepsilon(\phi)]\mathrm{d}x\mathrm{d}y}{\int_\Omega[1-H_\varepsilon(\phi)]\mathrm{d}x\mathrm{d}y},\\ \dfrac{\partial\phi}{\partial t}=\delta_\varepsilon(\phi)\left[\mu\nabla\cdot\dfrac{\nabla\phi}{|\nabla\phi|}-v-\lambda_o[I(x,y)-c_o]^2+\lambda_b[I(x,y)-c_b]^2\right],\\ \phi(0,x,y)=\phi_0(x,y)。\end{cases}\quad(7.8.3)$$

式中：

$$H_\varepsilon(z)=\frac{1}{2}\left[1+\frac{2}{\pi}\arctan\left(\frac{z}{\varepsilon}\right)\right],$$

$$\delta_\varepsilon(z)=\frac{1}{\pi}\frac{\varepsilon}{\varepsilon^2+z^2}。$$

从方程(7.8.3)可以看出，偏微分方程所涉及的图像函数 $I(x,y\nabla)$的数据是全图像定义域范围，而不是式(7.3.9)中的 $g(I(C))$函数，仅仅利用了轮廓线 C 所在位置的图像数据；另外，式(7.8.2)中的另外两个未知数 c_o,c_b 也定义在图像定义域内，具有全局特性。Chan-Vese 在其论文中阐述了该方法的一个非常显著的特点就是全局优化，因此可仅使用一条初始闭合轮廓线，就可以把带有内部空洞目标的内部全部检测出来，不需要为检测内部空洞而另外做特别处理，如图 7.14 所示。图 7.14(a)中要分割的目标是一个带有空洞的椭圆，图中的曲线是初始轮廓线。从图 7.14(b)可以看到，椭圆目标的内部空洞区域也分割出来了，整个分割过程仅使用了一条初始曲线；另外，初始曲线不需要完全位于区域的内部或外部，仍然可以正确地分割出目标和背景；这种方法还有一个显著的特点就是不依靠图像中的边界信息，因此，即使图像中的边界模糊或呈断续状，仍然可以获得理想的分割效果；最后，该方法还有消除噪声的作用，如图 7.14(c)所示，分割结束后所得的分割图像，就是带噪声图像的重建结果。

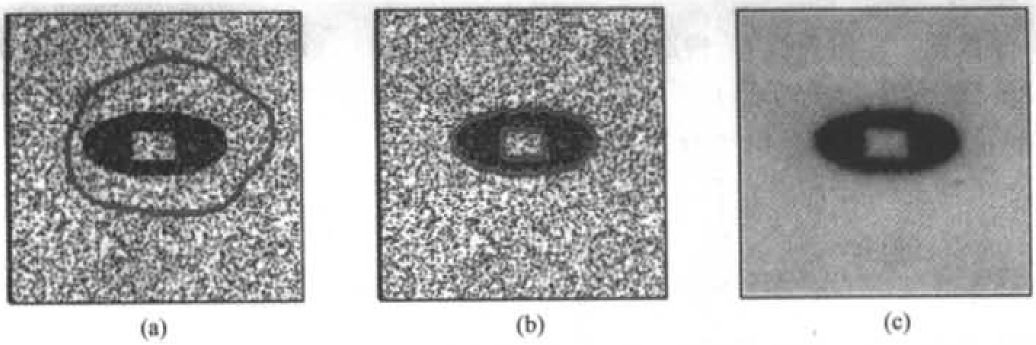


图 7.14 C-V 方法分割带有内部空洞的目标

(a) 待分割图像及初始活动轮廓线；(b) 最终分割结果；(c) 分割图像

7.8.2 C-V 分割方程存在的缺陷

C-V 方法的特点就是分割具有全局性，可以检测目标中间的空洞区域，然而，如果有多个待分割的目标，并且各个目标之间相距一定的距离，或者具有空洞区域目标的壁比较厚，则 C-V 方法常常不能得到正确的结果。

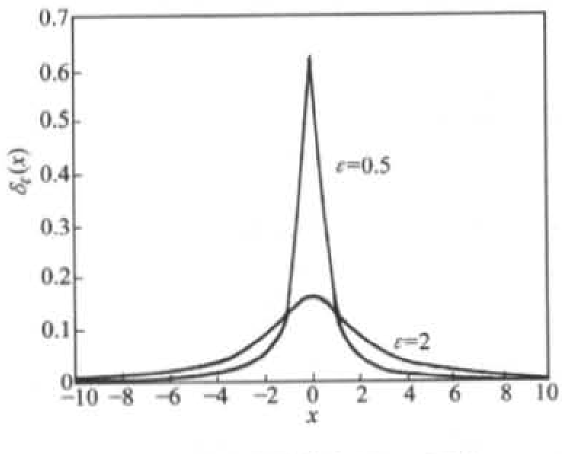


图 7.15 规则化的 Dirac 函数

实验表明，由于方程(7.8.3)中的 Dirac 函数 $\delta_\varepsilon(x)$狭窄的定义范围，很难保证仅靠一条初始轮廓线的演化，就将图像中所有的同质区域分割出来。C-V 方法中之所以能检测出带有空洞的目标的内部，也是因为采用了规则化的 Dirac 函数 $\delta_\varepsilon(x)$。该函数保证在图像定义域范围内，所有点的 $\delta_\varepsilon[\phi(x,y)]$值都趋于零的正值，从而不至于"埋没"空洞区域中的异质区域边界。然而，如果带空洞的目标比较"厚"，仍然不能稳定地检测出内部区域来。如图 7.15 所示，当 $\delta_\varepsilon(x)$中的参数 ε 比较小

时，方程(7.8.3)中其有效范围很窄，如图中的 $\varepsilon=0.5$ 的情况。可以看到，$\varepsilon>2$ 以后，$\delta_\varepsilon(x)$ 函数几乎为零了。这样，即使图像中某点 p 位于边界上，如果 p 点距离当前活动轮廓线比较远，即 $|\phi|$ 比较大，则 p 点的 ϕ 变化率几乎为零，该点的 ϕ 值就很难出现符号改变的情况，即出现活动轮廓线。这可以从图 7.16，图 7.17 中看出来。

图 7.16(a)中待分割的目标是一个带内部空洞区域的圆环，由于圆环的壁很厚，内部区域和初始轮廓线的距离远远超出了 $\delta_\varepsilon(x)$ 函数的有效范围，因此中心空洞仍然分割不出来，如图 7.16(b)的分割结果以及图 7.16(c)的分割图像所示。

而图 7.17(a)中，由于初始活动轮廓线距离下边和右边的两个几何图形较远，超出了 $\delta_\varepsilon(x)$ 的有效范围，因此，只有左边的几何图形分割出来，其他两个则没有被区分出。

另外，C-V 图像分割方程中，每次更新了水平集函数后，需要对水平集函数重新初始化为符号距离函数，以保持计算的稳定性。C-V 方法采用如下迭代方法重新初始化符号距离函数 ϕ：

$$\begin{cases}\psi_t = \text{sign}(\phi)(1-|\nabla\psi|),\\ \psi(0,\cdot)=\phi(t,\cdot)。\end{cases} \tag{7.8.4}$$

这种方法仅靠变形后的符号距离函数 ϕ 的符号来保证计算的准确性，而没有利用零水平集(Zero Level Set)确切位置信息，因此可能导致零水平集的位置发生移动。另外，这种方法在 Ω 域内的每点都要保证上式收敛，计算量也不小。

7.8.3　对 C-V 方程的改进——全局优化法

针对上述 C-V 分割方程中存在的问题，本文提出如下改进方案，以求快速分割图像，并提高能量函数式(7.8.2)的全局优化特性。

从图 7.15 中我们可以看出，如果增大 $\delta_\varepsilon(x)$ 函数的 ε 参数，则 $\delta_\varepsilon(x)$ 不为零的范围将扩大，如图中的 $\varepsilon=2$ 所示，则有可能检测出更多的图像边界。由于方程(7.8.3)中的 Dirac 函数抑制了非初始活动轮廓线附近的同质区域边界的检测，因此，如果将 $\delta_\varepsilon(x)$ 的有效范围扩大到全图像定义域 Ω，则整个图像定义域内的边界都可以检测出了。

基于上述思想，对方程(7.8.3)做如下改进，使得该方程的有效范围是整个图像定义域 Ω，这样，既可以检测出活动轮廓线 C 的内部区域边界，也可以同时检测出 C 外部区域，即背景中远离 C 的区域边界。

这里，我们将方程(7.8.3)中的 Dirac 函数 $\delta_\varepsilon[\phi(x,y)]$ 替换成 $|\nabla\phi(x,y)|$，即偏微分方程变为：

$$\frac{\partial\phi}{\partial t}=|\nabla\phi|\left[\mu\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}-v-\lambda_o[I(x,y)-c_o]^2+\lambda_b[I(x,y)-c_b]^2\right]。\tag{7.8.5}$$

方程(7.8.5)中的两个未知数 c_o 和 c_b 的计算同式(7.8.3)，即

$$c_o(\phi)=\frac{\int_\Omega I(x,y)H_\varepsilon(\phi)\mathrm{d}x\mathrm{d}y}{\int_\Omega H_\varepsilon(\phi)\mathrm{d}x\mathrm{d}y},\quad c_b(\phi)=\frac{\int_\Omega I(x,y)[1-H_\varepsilon(\phi)]\mathrm{d}x\mathrm{d}y}{\int_\Omega[1-H_\varepsilon(\phi)]\mathrm{d}x\mathrm{d}y}。$$

由于 ϕ 始终保持为符号距离函数，$|\nabla\phi|\approx1$，因此对于那些远离当前活动轮廓线的图像边界，其 ϕ 值变化率很大，且没有受到压抑，所以这些边界的 ϕ 值就可能出现符号翻转。这样，在这些边界的位置就会有活动轮廓线"冒"出来，方程(7.8.3)的有效范围可达整个图像定义域 Ω，而不仅仅是轮廓线 C 所在位置。由于没有 Dirac 函数的限制，方程(7.8.5)不仅可以检测出 C 的

内部边界，而且远离 C 的边界也可以检测出来，因此方程(7.8.5)比方程(7.8.3)有更好的全局优化特性。

7.8.4 方程的数值求解

为保证方程(7.8.5)数值解的稳定性，我们没有采用 C-V 方法中的隐式解法，而是采用如下显式有限差分数值解法：

设 h 为二维网格步长，$(x_i, y_i)=(ih, jh)$，$1\leqslant i,j\leqslant M$ 为格点坐标，则 $\phi_{i,j}^n=\phi(n\Delta t, x_i, y_i)$ 是关于 h 的对 $\phi(x,y)$ 在网格上的近似，这里有 $n\geqslant 0$，$\phi^0=\phi_0$，并设

$$D_{i,j}^{-x}=\phi_{i,j}-\phi_{i-1,j},\quad D_{i,j}^{+x}=\phi_{i+1,j}-\phi_{i,j},\quad D_{i,j}^{0x}=(\phi_{i+1,j}-\phi_{i-1,j})/2,$$
$$D_{i,j}^{-y}=\phi_{i,j}-\phi_{i,j-1},\quad D_{i,j}^{+y}=\phi_{i,j+1}-\phi_{i,j},\quad D_{i,j}^{0y}=(\phi_{i,j+1}-\phi_{i,j-1})/2。$$

根据 Osher-Sethian 求解水平集的“熵守恒”差分方法，得方程(7.8.5)的数值解表达式如下：

$$\phi_{i,j}^{n+1}=\phi_{i,j}^n+\Delta t\,[\max(F_{m\text{-}s},0)\,\nabla^{+}+\min(F_{m\text{-}s},0)\,\nabla^{-}$$
$$+\mu k_{i,j}[(D_{i,j}^{0y})^2+(D_{i,j}^{0x})^2]^{1/2},\tag{7.8.6}$$

式中：

$$\nabla^{+}=[\max(D_{i,j}^{-x},0)^2+\min(D_{i,j}^{+x},0)^2+\max(D_{i,j}^{-y},0)^2+\min(D_{i,j}^{+y},0)^2]^{1/2}$$
$$\nabla^{-}=[\min(D_{i,j}^{-x},0)^2+\max(D_{i,j}^{+x},0)^2+\min(D_{i,j}^{-y},0)^2+\max(D_{i,j}^{+y},0)^2]^{1/2}$$

$k_{i,j}$是水平集函数在(i,j)的曲率，由下式定义：

$$k=\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}=\frac{\phi_{xx}\phi_y^2-2\phi_{xy}\phi_x\phi_y+\phi_{yy}\phi_x^2}{(\phi_x^2+\phi_y^2)^{3/2}},$$

可以由中点差分 $D_{i,j}^{0x}$ 和 $D_{i,j}^{0y}$ 近似计算出来。

$F_{m\text{-}s}$是由 Mumford-Shah 分割模型所定义的项，

$$F_{m\text{-}s}(x,y)=-v-\lambda_o[I(x,y)-c_o]^2+\lambda_b[I(x,y)-c_b]^2。$$

由于方程(7.8.5)需要求解能量函数式(7.8.2)的全局解，因此不能使用通常局部求解水平集的方法——窄带法(Narrow Band)，需要在整个定义域 Ω 内更新水平集函数，计算量大。不过，方程(7.8.5)是全局优化偏微分方程，从后面的实验可以看到，只需要数量很少的几次迭代，就可以获得理想的分割效果。其实，真正影响水平集计算量的步骤是每次更新水平集函数后，将水平集函数重新初始化为符号距离函数的计算。本文采用了前面所述的最近点映射构造符号距离函数的方法，该方法仅需在 Ω 范围内做四次扫描和简单的计算，就可以准确地构造出距离函数 ϕ，计算量仅为 $O(N)$。

实际计算中，为避免不必要的迭代计算，我们检查如下迭代收敛的判断条件：

$$Q=\frac{\sum\limits_{|\phi_{i,j}^n|<h}|\phi_{i,j}^{n+1}-\phi_{i,j}^n|}{M}\leqslant h^2\Delta t。\tag{7.8.7}$$

式中：M 是满足$|\phi_{i,j}^n|<h$ 的网格点数目。

总结一下，改进的具有全局优化特性的 M-S 分割方法计算步骤如下：

① 初始化：定义初始活动轮廓线 C_o，并根据 C_o，利用前述快速符号距离函数构造方法，构造 ϕ_o；

② 根据当前 ϕ^n，计算 $c_o(\phi^n)$，$c_b(\phi^n)$和 $L(\phi^n)$；

③ 根据当前 c_o、c_b 和 L，利用迭代式(7.8.6)，计算 ϕ^{n+1}，并更新活动轮廓线 C；

④ 检查迭代是否收敛，如果收敛，则停止计算，否则，转步骤②，继续计算。

7.9　改进 M-S 分割模型方法实验结果

下面用 C-V 算法和本文方法对分割图像实验做比较，比较的指标是迭代次数、计算时间以及最终分割结果。实验中，C-V 方法以及本文改进的 C-V 分割方法都采用本文提出的快速符号距离函数产生算法，以提高计算速度和稳定性。比较的两种方法都采用了相同的参数。所有的实验中，两种方法的时间步长 $\Delta t=0.1$，网格步长 $h=1$，M-S 模型参数：$\lambda_o=\lambda_b=1$，$v=0$，$\varepsilon=1$。

图 7.16 是采用 C-V 方法和本文改进的全局分割方法分割的结果比较。从图可以看到，由于圆环的壁比较厚，C-V 方法不能分割出圆环中的空洞区域；而文本改进的分割方法可将所有的几何图形全部分割出来，整个分割过程仅迭代了 3 次，耗时 0.90s；而 C-V 方法得到结果需要迭代 4 次，结合本文的快速距离函数构造方法，耗时 1.2s。从图 7.16(e)还可以看出，基于 M-S 分割模型的方法不仅可以分割目标，而且还可以重建被噪声污染的图像。

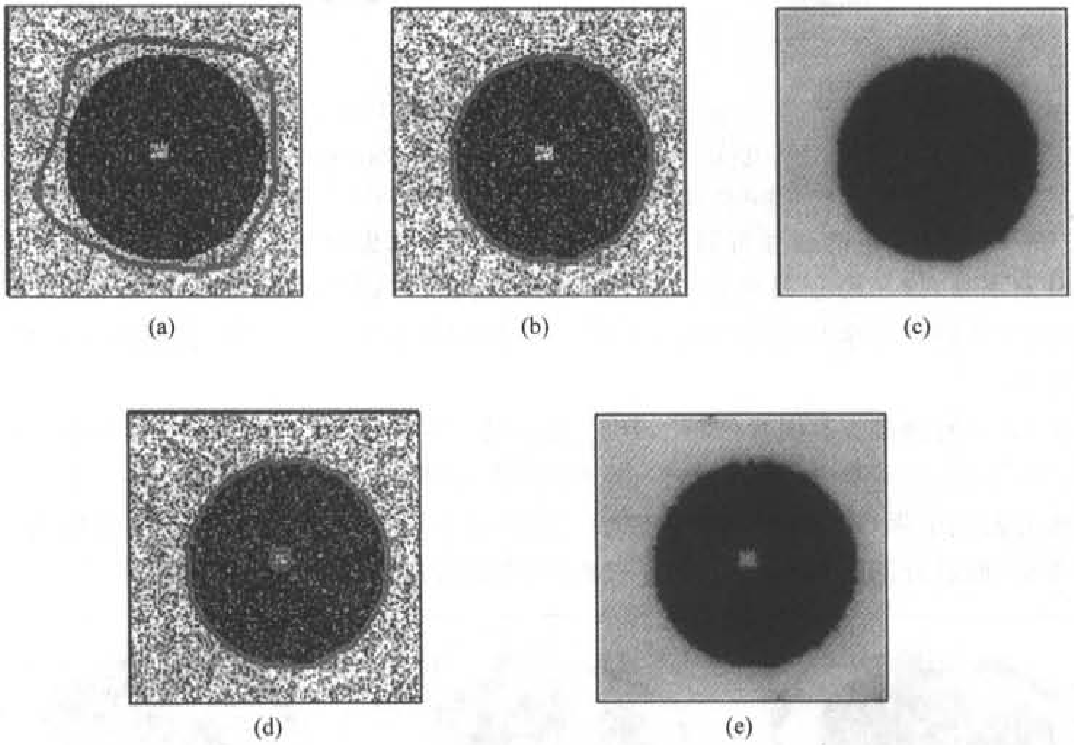


图 7.16　C-V 方法和本文改进方法分割结果对比

(a)待分割图像(128×128)和初始轮廓线；(b)C-V 方法最终分割结果；
(c) C-V 分割图像；(d) 改进方法分割结果；(e) 改进方法分割图像

图 7.17 是对合成带噪声图像(大小 130×130)的分割实验。其中，图 7.17(a)是原始图像和初始轮廓线，图中有三个几何图形，其中两个有空心区域，初始轮廓线包含左上角的目标图形。图 7.17(b)是利用 C-V 方法分割的结果，共迭代 6 次，计算时间是 1.125s；图 7.17(c)是分割重建的结果，可以看到，右边的另外两个几何图形没有重建出来；图 7.17(d)是本文改进方法的分割结果，迭代 3 次，耗时 0.561s；图 7.17(e)是分割重建的结果。

从实验结果可以看出，本文方法不仅迭代次数减少了，而且分割结果比 C-V 方法好。图

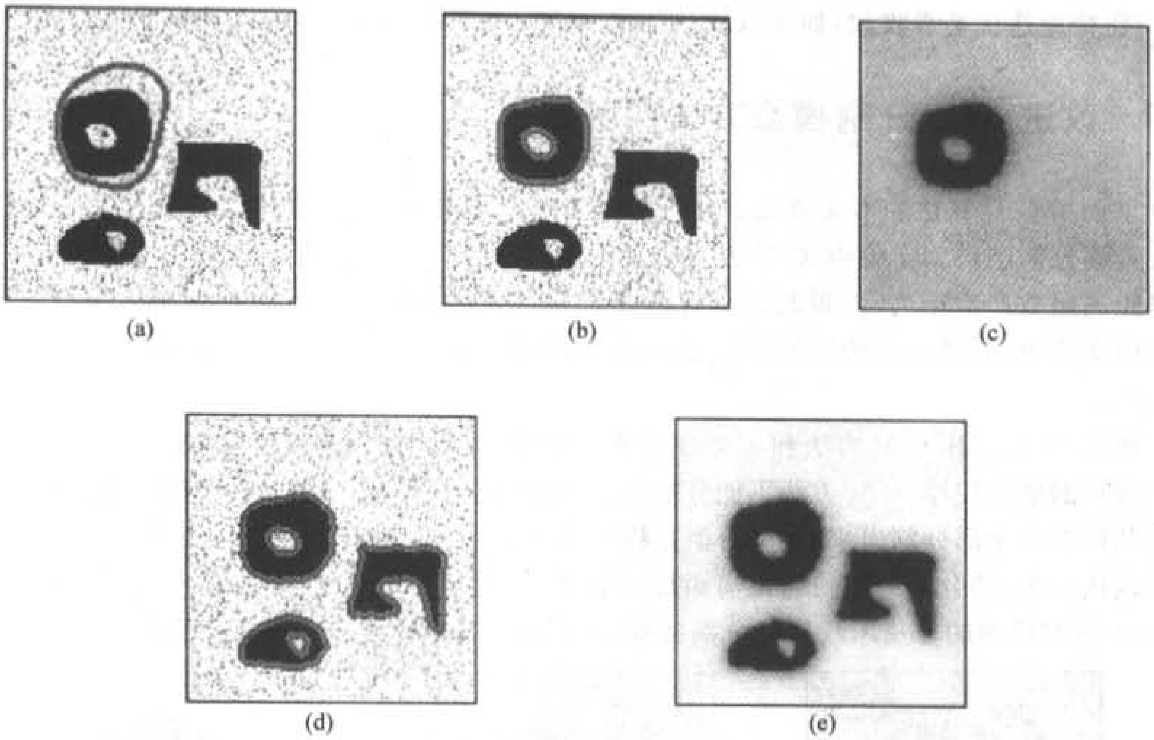


图 7.17　带噪声合成图像的分割

(a)初始化(130×130)；(b)C-V 算法分割结果(6 次，1.125s)；

(c) C-V 分割重建图像；(d) 改进方法分割结果(3 次，0.561s)；(e) 改进方法分割重建图像

7.17(c)中，只有左上角的目标得到了完好分割(内部区域也分割出来了)，但是另外两个目标就没有分割出来，这是由于另外两个目标距离初始轮廓线较远，超出了 Dirac 函数的有效范围；而本文方法得到了全局的分割效果(如图 7.17(e)所示)，三个目标(包括内部区域)均很好地分割出来。

图 7.18 是对胸腔 CT 图片(217×185)的分割。其中图 7.18(a)是原始图像和初始轮廓线；图 7.18(b)是 C-V 算法分割的结果(迭代 30 次，耗时 14.921s)；图 7.18(c)是本文改进的 C-V 方法分割的结果，迭代仅 4 次就收敛了，耗时仅 2.033s。可以看到，本文改进的方法不仅收敛非常快，而且分割的结果比 C-V 方法满意，大部分的骨骼区域都划分出来了。

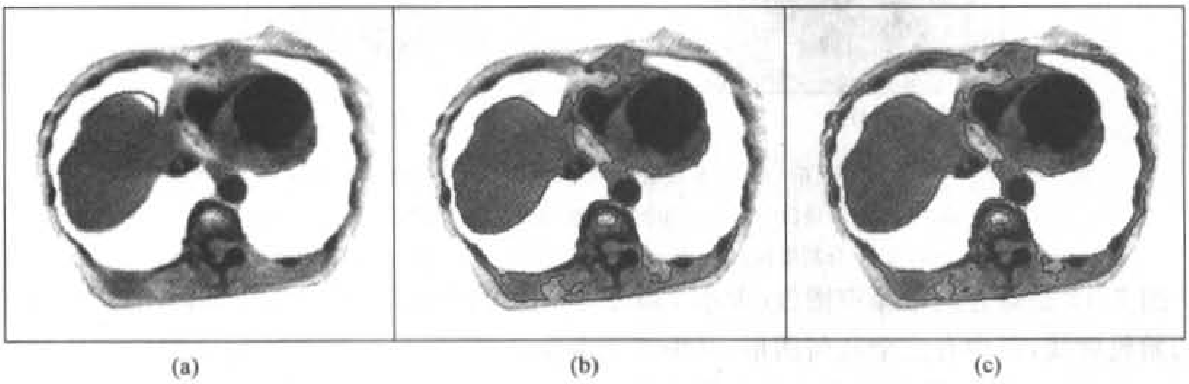
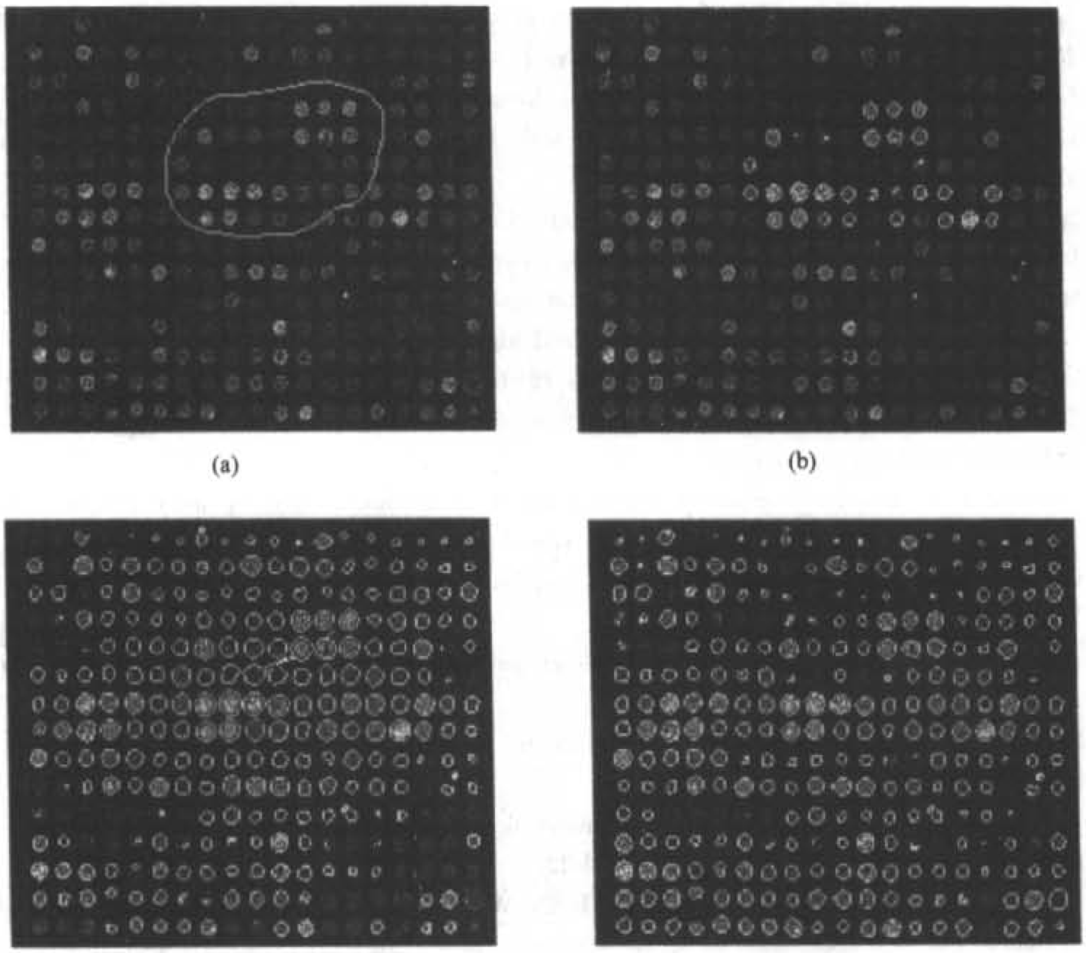


图 7.18　胸腔 CT 图片的分割

(a)初始化(217×185)；(b)C-V 算法分割结果(30 次，14.921s)；(c) 本文方法分割结果(4 次，2.033s)

图 7.19 是对生物芯片图像中探针点阵的分割结果。生物芯片中探针点(见图 7.19(a)中的圆点)的亮度和颜色是生物化学反应强度的表征。要提取每个探针点的这些信息,首要任务是定位探针点位置。但是,由于探针点亮度不均匀,而且背景噪声很大,因此常规基于边缘分割的方法很难获得理想的分割结果。由于 M-S 模型是基于空间和灰度的信息模型,因此,很适合作探针点的分割应用。图 7.19(a)就是一幅灰度化后的芯片图像和初始轮廓线,为了突出本文方法分割的全局性,我们故意将轮廓线仅包含一部分探针点。图 7.19(b)是基于图 7.19(a)初始化条件下,用 C-V 方法做的分割。可以看出,C-V 方法只能分割出初始轮廓线内部的探针点以及周围很窄范围内的探针点,不能将全图的探针点都分割出来。如果想作全图的探针点分割,必须使初始轮廓线包含全图。基于这种方式的初始化条件下,要收敛需要很多次迭代,很耗时。图 7.19(c)、(d)是利用本文改进方法做的分割结果,其中图 7.19(c)是仅迭代了一次的结果,仅仅迭代了一次,就分割出了很多探针点;图 7.19(d)是最终的分割结果,仅 5 次迭代,耗时不到 4s。而使用基于圆心度的方法定位探针点位置,则同样大小的图像需要 5min 左右,结果还不一定稳定。这再一次表明了本文方法的全局特征和快速收敛特性。



(a) (b) (c) (d)

图 7.19 生物芯片探针矩阵点的分割

(a)初始化(450×400);(b)C-V 分割结果(28 次,59.045s);

(c) 本文方法中间结果(1 次,2.211s);(d) 本文方法最终结果(5 次,11.056s)

参 考 文 献

[1] Malladi R, Sethian J A, Vemuri B C. Shape modeling with front propagation: a Level Set approach, IEEE Trans. On PAMI, 1995, 17 (2): pp.158-pp.175

[2] Alvarez L, Lions P L, Morel J M. Image selective smoothing and edge detection by nonlinear diffusion Ⅱ. SIAM Journal on Numerical Analysis, 1992, 29(3): pp.845-pp.866

[3] Paragios N, Deriche R. Geodesic active contours for supervised textures Segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, Colorado, USA, 1999

[4] Paragios N D. Geodesic active contours and Level Sets for the detection and tracking of moving objects. IEEE Transactions On Pattern Analysis and Machine Intelligence, 2000, 22 (3): pp.266-pp.280

[5] Faugeras O, Keriven R. Variational principles, surface evolution, PDE's Level Set methods, and the stereo problem. IEEE Transactions on Image Processing, 1998, 7(3): pp.336-pp.344

[6] Bertalmio M, Sapiro G, Caselles V, et al. Image inpainting. Computer Graphics (SIGGRAPH), pp. 417-pp.424, New Orleans, July 2000

[7] Kimia B B. Conservation laws and a theory of shape, Ph. D thesis, McGill Centre for Intelligent Machines, McGill Univ., Montreal, Canada, 1990

[8] Grayson M A. Shortening embedded curves, Annals of Mathematics, 1989, 129: pp.71-pp.111

[9] Caselles V, Kimmel R, Sapiro G. Geodesic active contours. International Journal of Computer Vision, 1997, 22(1): pp.61-pp.79

[10] Sethian J A, Curvature and evolution of fronts. Commun. Math. Phys., 1985, 101: pp.487-pp.499

[11] Osher S, Sethian J A. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton -Jacobi formulations. Journal of Computational Physics, 1988, 79: pp.12-pp.49

[12] Sethian J A. A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed. J. Differential Geometry, 1989, 31: pp.131-pp.161

[13] Sethian J A. An analysis of flame propagation. Ph. D Thesis, Dept. of Mathematics, University of California, Berkeley, CA, 1982

[14] Sethian J A. Level Set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge University Press, 1999

[15] Chop D. Computing minimal surfaces via Level Set curvature flow. Journal of Computational Physics, 1993, 106: pp.77-pp.91

[16] Adalsteinsson D, Sethian J A. A fast level set method for propagating interface. Journal of Computational Physics, 1995, 118: pp.269-pp.277

[17] Sethian J A. A fast marching level set method for monotonically advancing fronts. In Proc. Nat. Ac. Science, 1996, 93: pp.1591-pp.1694

[18] Adalsteinsson D, Sethian J A. The fast construction of extension velocities in Level Set methods. Journal of Computational Physics, 1999, 148: 2-22

[19] Barrow H G, Tenenbaum J M, Bolles R C, Wolf H C. Parametric correspondence and chamfer matching two new techniques for image matching. proc. 5th Joint Conf. Artificial Intelligence, Cambridge, pp.659-pp.663, 1977

[20] Leventon M E. Statistical models in medical image analysis. Ph. D. Thesis, Dept. of E&E and CS, MIT, Massachusetts. CA, 2000

[21] Tsai Y H. Rapid and accurate computation of the distance functioon using grids. UCLA CAM Report 00-36

[22] Xu C, Pham D L, Prince J L. Medical Image Segmentation Using Deformable Models. SPIE Handbook

on Medical Imaging — Volume Ⅲ: Medical Image Analysis, edited by J. M. Fitzpatrick and M. Sonka, May 2000

[23] 刘儒勋,王志峰. 数值模拟方法和运动界面追踪. 合肥:中国科技大学出版社,2001

[24] 李俊,杨新等. 基于 Mumford-Shah 准则的快速 Level Set 图像分割方法及其应用. 北京:计算机学报, 2002, 11

[25] 王峥,杨新等. 水平集方法中曲线跟踪算法的改进. 上海:上海交通大学学报,2002,36(12)

[26] 王峥,杨新等. 基于水平集人机交互模型的医学图像分割. 合肥:模式识别与人工智能,2002,15(4)